# Arduino analogRead analogWrite

## digitalWrite

Escriba un valor HIGH o LOW en un pin digital. Si el pin ha sido configurado como OUTPUT con pinMode(), su voltaje se establecerá en el valor correspondiente: 5V (o 3.3V en placas de 3.3V) para HIGH, 0V (tierra) para LOW. Si el pin está configurado como INPUT, digitalWrite() habilitará (HIGH) o deshabilitará (LOW) el pullup interno en el pin de entrada. Se recomienda establecer pinMode() en INPUT_PULLUP para habilitar la resistencia pull-up interna. Si no configura pinMode() en OUTPUT y conecta un LED a un pin, cuando llame a digitalWrite (HIGH), el LED puede aparecer tenue. Sin establecer explícitamente pinMode(), digitalWrite() habrá habilitado la resistencia pull-up interna, que actúa como una gran resistencia limitadora de corriente.

## digitalRead

Lee el valor de un pin digital especificado, ya sea HIGH o LOW.

## analogRead

Lee el valor del pin analógico especificado. Las placas Arduino contienen un convertidor analógico a digital multicanal de 10 bits. Esto significa que mapeará los voltajes de entrada entre 0 y el voltaje de operación (5V o 3.3V) en valores enteros entre 0 y 1023. En un Arduino UNO, por ejemplo, esto produce una resolución entre lecturas de: 5 voltios / 1024 unidades o , 0,0049 voltios (4,9 mV) por unidad.

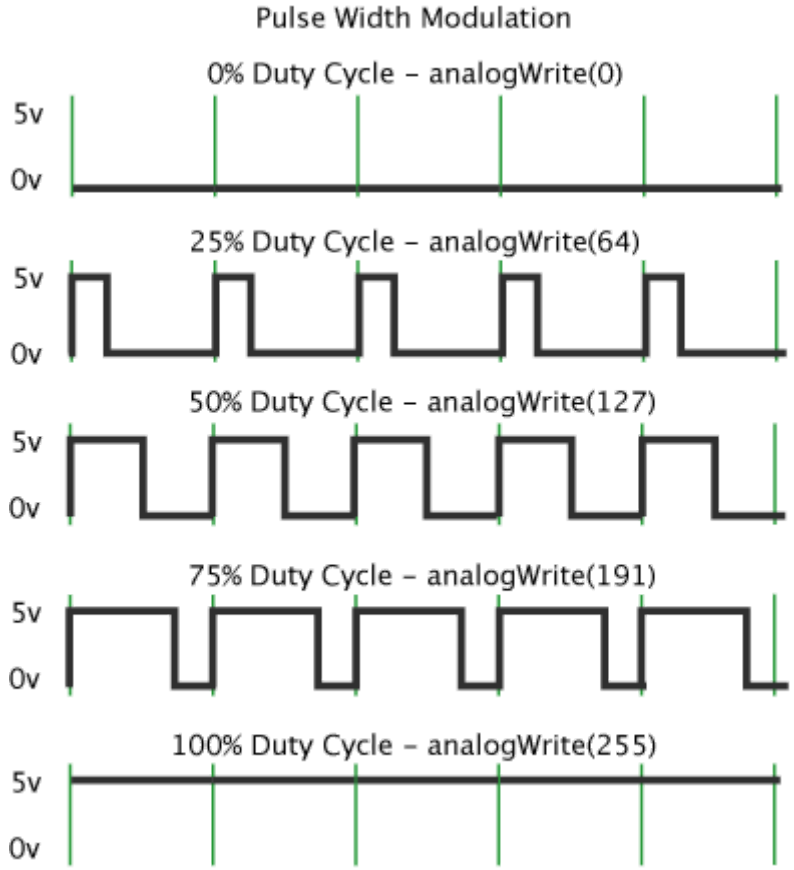
## Tilde de la eñe

La virgulilla es un signo ortográfico en forma de coma, onda o trazo. La virgulilla generalmente es identificada como la tilde de la eñe o virgulilla de la eñe (~), aunque la Real Academia Española acepta también como ejemplos de virgulilla el apóstrofo (’), la cedilla (¸), y el acento agudo (´). Las palabras «tilde» y «virgulilla» se pueden referir a cualquier trazo, sin embargo, el contexto puede indicar si específicamente se alude o no al signo aquí referido, o sea a (~).

## PWM

La modulación de ancho de pulso, o PWM, es una técnica para obtener resultados analógicos con medios digitales. El control digital se utiliza para crear una onda cuadrada, una señal que cambia entre encendido y apagado. Este patrón de encendido y apagado puede simular voltajes entre el Vcc completo de la placa (p. ej., 5 V en UNO, 3,3 V en una

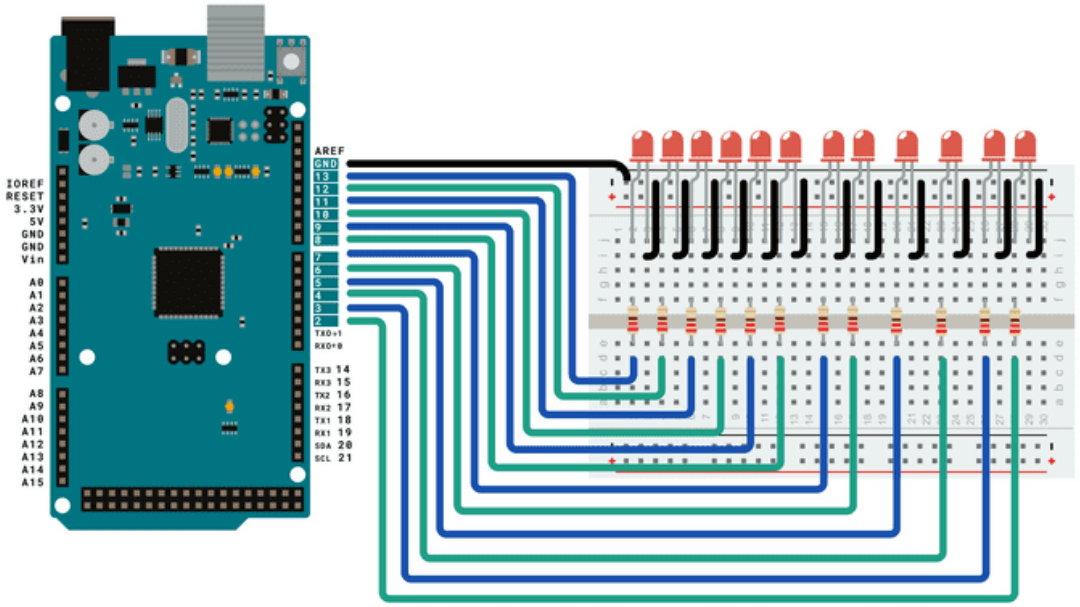
placa MKR) y apagado (0 voltios) al cambiar la parte del tiempo que pasa la señal en comparación con el tiempo que la señal pasa apagada. La duración de "a tiempo" se denomina ancho de pulso. Para obtener valores analógicos variables, cambia o modula ese ancho de pulso. Si repite este patrón de encendido y apagado lo suficientemente rápido con un LED, por ejemplo, el resultado es como si la señal fuera un voltaje constante entre 0 y Vcc que controla el brillo del LED. En Arduino UNO los pines con esta característica su número está antecedido por una tilde de la eñe.



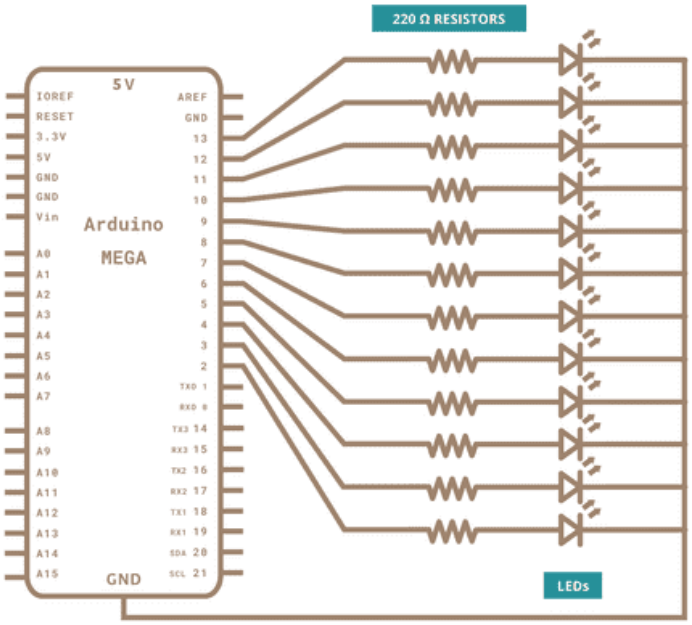

**analogWrite**

Escribe un valor analógico (onda PWM) en un pin. Se puede usar para encender un LED con diferentes brillos o impulsar un motor a varias velocidades. Después de una llamada a analogWrite(), el pin generará una onda rectangular constante del ciclo de trabajo especificado hasta la próxima llamada a analogWrite() (o una llamada a digitalRead() o digitalWrite()) en el mismo pin. Los pines con esta característica son los números 3, 5, 6, 9, 10 y 11.

## Circuito


IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
TX→1
RX←0
TX3 14
RX3 15
TX2 16
RX2 17
TX1 18
RX1 19
SDA 20
SCL 21

## Esquema


220 Ω RESISTORS
5V
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
Arduino
MEGA
GND
AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
TXD 1
RXD 0
TX3 14
RX3 15
TX2 16
RX2 17
TX1 18
RX1 19
SDA 20
SCL 21
LEDs

**AnalogWriteMega.ino**

```
/*
  Mega analogWrite() test

  This sketch fades LEDs up and down one at a time on digital pins 2 through 13.
  This sketch was written for the Arduino Mega, and will not work on previous boards.

  The circuit:
  * LEDs attached from pins 2 through 13 to ground.

  created 8 Feb 2009
  by Tom Igoe

  This example code is in the public domain.

 */
// These constants won't change.  They're used to give names to the pins used:
const int lowestPin = 9; // originalmente 2
const int highestPin = 11; // originalmente 13


void setup() {
  // set pins 2 through 13 as outputs:
  for (int thisPin =lowestPin; thisPin <= highestPin; thisPin++) {
    pinMode(thisPin, OUTPUT);
  }
}

void loop() {
  // iterate over the pins:
  for (int thisPin =lowestPin; thisPin <= highestPin; thisPin++) {
    // fade the LED on thisPin from off to brightest:
    for (int brightness = 0; brightness < 255; brightness++) {
      analogWrite(thisPin, brightness);
      delay(2);
    }
    // fade the LED on thisPin from brithstest to off:
    for (int brightness = 255; brightness >= 0; brightness--) {
      analogWrite(thisPin, brightness);
      delay(2);
    }
    // pause between LEDs:
    delay(100);
  }
}
```

**Consigna**

Subir a la página de la EIS, un documento con carátula, una breve explicación de las funciones setup, loop y serialEvent; cuyo nombre sea el/los apellido/s completo/s seguido de la/s inicial/es de su/s nombre/s, en formato .pdf.

**Cibergrafía**

digitalWrite
https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/

digitalRead
https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/

analogRead
https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/

Tilde de la eñe
https://es.wikipedia.org/wiki/Virgulilla

PWM
https://docs.arduino.cc/learn/microcontrollers/analog-output

analogWrite
https://www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/

Circuito
https://www.arduino.cc/en/Tutorial/BuiltInExamples/AnalogWriteMega

Esquema
https://www.arduino.cc/en/Tutorial/BuiltInExamples/AnalogWriteMega

AnalogWriteMega.ino
https://www.arduino.cc/en/Tutorial/BuiltInExamples/AnalogWriteMega