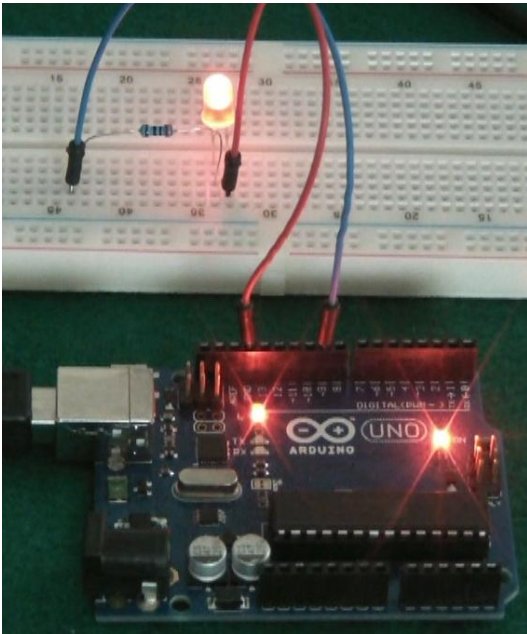


## 03 - effetto fade, gestione tecnica PWM - Fade effect, PWM technical management (some notes at end of this section)



PWM (Pulse Width Modulation) e' la tecnica con la quale Arduino utilizza in maniera analogica una porta digitale di output. Arduino dispone di 6 porte digitali (la 3, 5, 6, 9, 10 e 11) che, tramite l'istruzione:

*analogWrite (porta, parametro);*

possono fornire tensioni variabili e quindi pilotare in modalita' analogica gli attuatori ad esse collegate. Valorizzando *parametro* con un numero da 0 a 255 Arduino fornira' alla *porta* di output una tensione variabile da 0 a 5 volt, proporzionale al parametro. Con *parametro* 0 fornira' una tensione pari a 0 mentre con *parametro* 255 fornira' una tensione di 5 volt

In questo esempio Arduino gestisce la luminosita' di un led portandola progressivamente dal minimo al massimo.

**Nota:** Questo esercizio e questa nota sono parte di una serie che vede protagonisti Arduino ed alcuni dei componenti ad esso collegabili. Per la maggior parte degli esercizi e' anche disponibile un filmato su youtube.

- [Esercizi facenti parte della raccolta](#)
- [Filmati presenti su youtube](#)
- [Informazioni su arduino e sui componenti collegabili \(PDF scaricato nell'area di download\)](#)
- [Breve manuale di programmazione \(PDF scaricato nell'area di download\)](#)

Per eventuali chiarimenti o suggerimenti sul contenuto di questa scheda scrivere a [giocarduino@libero.it](mailto:giocarduino@libero.it)

**Here some notes about this project, translated by google translator**



PWM (Pulse Width Modulation) is the technique with which Arduino uses a digital output pin in an analog manner. Arduino has 6 digital pins (3, 5, 6, 9, 10 and 11) that, through this instruction:

*analogWrite (pinnumebr, parameter);*

They can provide varying voltages and drive, in analog mode, the connected actuators. *Parameter* can be a number from 0 to 255 by which Arduino will send to the output pin a variable voltage from 0 to 5 volts, proportional to the parameter. With parameter 0 will provide a voltage 0 while with parameter 255 will provide a voltage of 5 volts.

In this example Arduino controls the brightness of a LED gradually bringing it from minimum to maximum.

**Note:** This project and this note is part of a series that sees, as main characters, Arduino and some of connectable components. For most projects there is also a video on youtube.

- [Projects collection](#)
- [Movies on youtube](#)

## Arduino: effetto fade, gestione tecnica PWM - fade effect, PWM technical management

- [About Arduino and components \(italian; pdf will be downloaded in your download area\)](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

For any questions or suggestions about this note (and on its english translation), please write to [giocarduino@libero.it](mailto:giocarduino@libero.it) (simple words and short sentences, please)

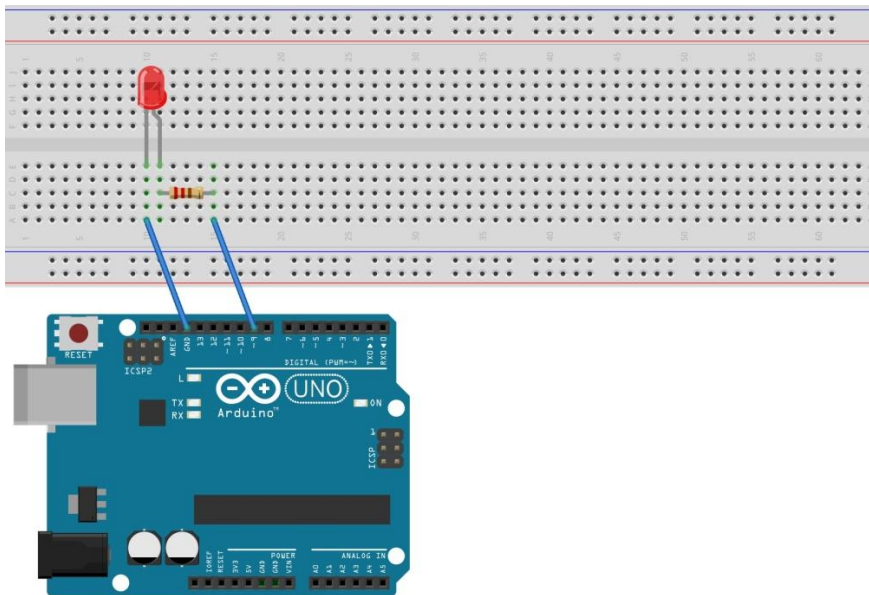
For any questions or suggestions about this note (and on its english translation), please write to [giocarduino@libero.it](mailto:giocarduino@libero.it) (simple words and short sentences, please)

For any questions or suggestions about this note (and on its english translation), please write to [giocarduino@libero.it](mailto:giocarduino@libero.it) (simple words and short sentences, please)

## Materiali

- una breadboard
- una resistenza da 220 ohm
- un led

## Schema



fritzing

## Programma

```
/* Attenzione: facendo il copia/incolla dal PDF all'IDE si perde la formattazione del testo. Per
* rendere piu' facilmente leggibile il programma e' opportuno formattarlo subito dopo il
* trasferimento nell'IDE, premendo CTRL+T.
*
* PWM (pulse width modulation) e' una tecnica in grado di produrre segnali di tipo analogico su di
* una porta digitale In questo esercizio si agisce sulla tensione in uscita dalla porta 9, facendola
* variare progressivamente da 0 a 5 volt e quindi, sempre progressivamente, da 5 a 0 volt
* Per fare cio' si agisce sul secondo parametro dell'istruzione "analogWrite" facendogli assumere
* valori che vanno progressivamente da 0 a 255. Il valore zero provoca l'uscita di un segnale con la
* tensione minima (e cioe' 0 volt) mentre 255 provoca l'uscita di un segnale con la tensione massima
* (e cioe' 5 volt) Il circuito prevede un diodo con il catodo collegato alla terra (gnd) e l'anodo
* collegato ad una resistenza da 220 ohm a sua volta collegata alla porta 9 (che, formalmente, e'
* una porta digitale)
*
* -----
* Warning: cut&paste from PDF to IDE loses formatting. to restore it press CTRL + T.
* PWM (pulse width modulation) is a technique capable of producing analog signals on a digital pin.
* In this project, one acts on the voltage on pin 9, progressively varying from 0 to 5 volts, and
```

## Arduino: effetto fade, gestione tecnica PWM - fade effect, PWM technical management

```
* then, again progressively, from 5 to 0 volts. To do this you act on the second parameter of the
* "analogWrite" instruction, giving it progressive values, ranging from 0 to 255. A value of zero
* causes a signal output with the minimum voltage (and that means 0 volts) while 255 causes the
* output of a signal with the maximum voltage (5 volts). The circuit includes a lighting diode with
* the cathode connected to ground and the anode connected to a resistance of 220 ohms in turn
* connected to pin 9 (which, formally, is a digital pin)
*-----
*/
int lucentezza = 0;           // definisce la variabile "lucentezza" con valore iniziale 0
int incremento = 5;          // definisce la variabile "incremento" con valore 5
//
void setup ()
{
  pinMode (9, OUTPUT);       // definisce la porta 9 come porta di output
}
//
void loop()
{
  analogWrite (9, lucentezza); // invia alla porta 9 una tensione pari a (5/255) * lucentezza
  lucentezza = lucentezza + incremento; // modifica il valore di lucentezza
  if (lucentezza == 0 || lucentezza == 255) /* il simbolo || e' un operatore booleano che
  rappresenta la "or" mentre == e' il simbolo convenzionalmente utilizzato per verificare la
  condizione di uguaglianza. Non viene utilizzato il normale simbolo = poiche' il programma lo
  interpreterebbe come l'assegnazione di un valore es. lucentezza = 0 verrebbe interpretato
  come "assegna 0 alla variabile lucentezza" - The combination " || " is a Boolean operator that
  represents the "or", while " == " symbol is conventionally used to indicate a condition of
  equality. Is not used the normal symbol " = " because the program interprets it as an assigning
  value. Example: lucentezza = 0 would be interpreted as "assigns 0 to the variable lucentezza
  (brightness)" */
  {
    incremento = incremento *-1; /* se la lucentezza ha raggiunto gli estremi ( 0 oppure 255) cambia
    segno all'incremento (lo si moltiplica per -1 e quindi se era 5 lo fa diventare -5 e viceversa) - if
    the brightness has reached the extremes (0 or 255) changes sign to the increase (multiplies it by -1
    and if it was 5, becomes -5, and vice versa) */
  }
  delay (50);                //attende 50 millisecondi prima di ricominciare il ciclo
}

/* Nota: l'utilizzo di un ciclo di for avrebbe reso piu' semplice la scrittura del programma, ma in
questi primi esercizi sono state utilizzate solo istruzioni di base, per consentire al lettore una
piu' facile comprensione del programma e delle modalita' di utilizzo delle istruzioni - Note: the
use of a "for" cycle would make it easier write the program, but in these first projects we have
used only the basic instructions, to allow the reader an easier understand program and instructions
*/
```