

organized by  
VERONAFIERE

MODEL  
EXPO  
ITALY

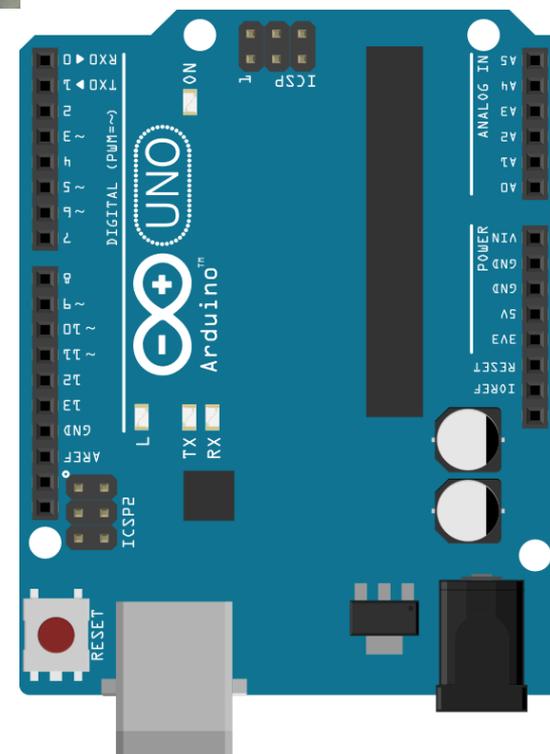
VERONA 21/22 FEBBRAIO 2015  
LA FIERA DEL MODELLISMO PIÙ IMPORTANTE IN ITALIA

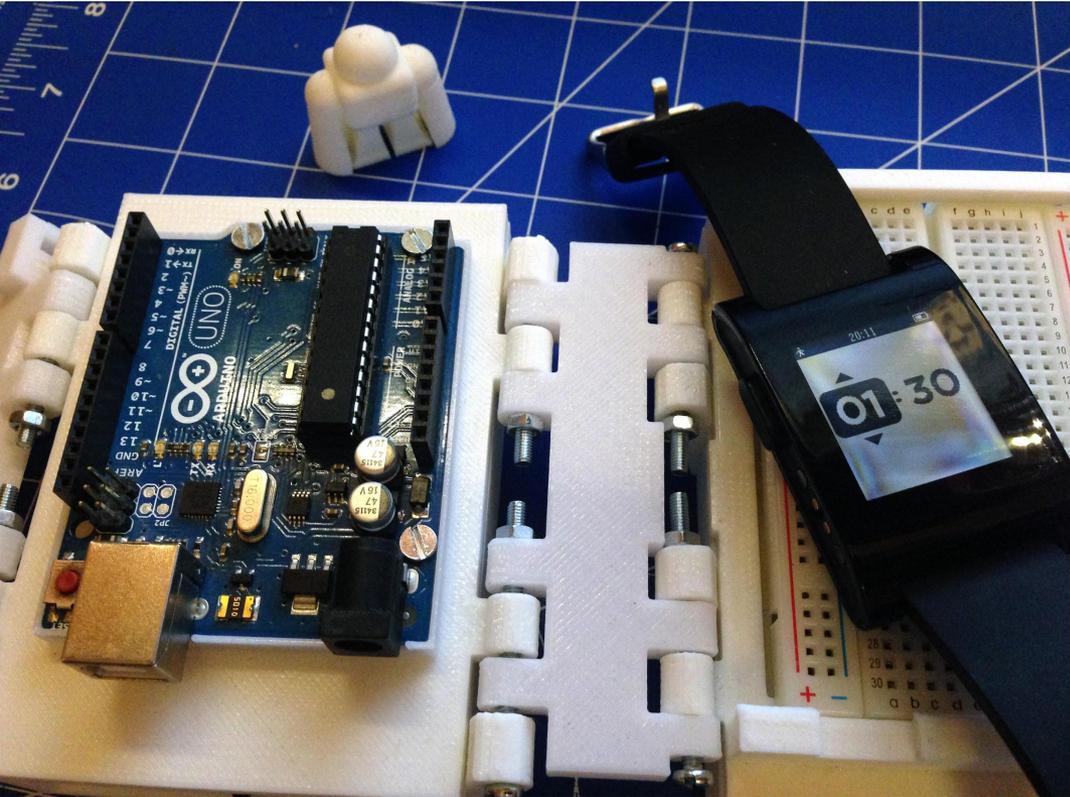


# Introduzione all'uso di Arduino

Prof. Michele Maffucci

Model Expo Italy - Verona 21/22 febbraio 2015





# Comprendere l'universo Arduino in 90 minuti

**Questa presentazione, nata in occasione del Workshop organizzato dal FabLab di Brescia per il ModelExpo 2015, vuole essere un'introduzione all'uso di Arduino e ben si adatta a tutti i maker, studenti ed adulti, che per passione nell'elettronica necessitano di un'introduzione che potrà poi portare, dopo un'approfondita sperimentazione, a sviluppare progetti di automazione anche nel campo del modellismo.**

**In questa esposizione troverete le linee guida che potranno condurvi alla strutturazione di un vostro percorso di apprendimento in modo che possiate raggiungere le competenze necessarie per lo sviluppo dei vostri progetti.**

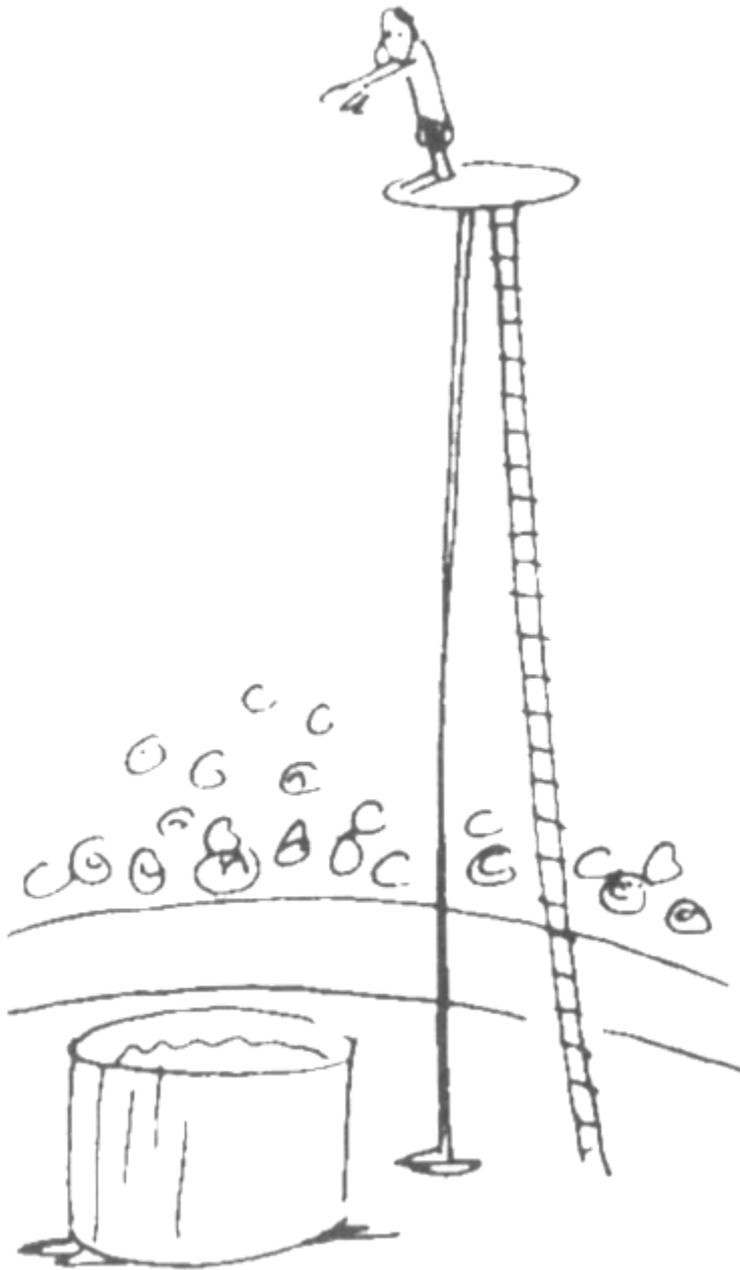
I contenuti esposti sono un estratto di quanto scritto in occasione dei miei corsi presso scuole, centri di formazione e FabLab, ulteriori approfondimenti e risorse a questa presentazione possono essere trovate sul mio sito personale dove potrete seguire corsi introduttivi all'uso di Arduino

<http://www.maffucci.it/area-studenti/arduino/>

*Il codice e le slide utilizzate sono suscettibili di variazioni/correzioni che potranno essere fatte in ogni momento.*

## Argomenti

- Come incominciare
- Impostazione metodologica
- Cos'è Arduino?
- Perché usare Arduino?
- Cosa posso fare con Arduino
- Competenze di base - dove imparo l'elettronica e l'elettrotecnica
  - siti
  - libri per chi incomincia
  - libri per approfondire
- Competenze di base - dove imparo a programmare
  - libri - esercizi - videocorsi
- Dove imparo l'Alfabeto di Arduino
  - libri
  - corsi on-line
- Applicativi essenziali per lo studio e la progettazione
- Universo Arduino
- La scheda Arduino
- Terminologia essenziale
- Il software Arduino
- Comunicare con Arduino
- Programmazione
- Il primo programma
- Prodotti Arduino
- Quale scheda scegliere
- Costruirsi una scheda Arduino
- Estendere le funzionalità con gli shield
- Pensieri



# Come incominciare

*non preoccuparti non è  
così complicato :-)*

# ***Problema***

- 1** *non conosco l'elettronica*
- 2** *non ho mai programmato*

***“Posso farcela?”***

***Sì !***

*vediamo come...*

# Impostazione metodologica



# L'agire del maker alle prime armi che opera con Arduino...



*correre per raggiungere l'obiettivo*

**ERRORE!**

il rischio, per chi non ha mai affrontato argomenti che riguardano **l'elettronica** è:



*frustrazione ed abbandono*

**NON DEVE ACCADERE!**

# ***Obiettivi***

- 1** *costruire la competenza*
- 2** *trovare la soluzione al proprio problema*

# Atteggiamento sbagliato



*Domana: “...ho un albergo, voglio usare Arduino per rilevare la temperatura nelle camere, la temperatura esterna e comandare la centrale termica dell'albergo ed inoltre mi piacerebbe un controllo via cellulare... ho poche competenze in elettronica e conosco poco Arduino... come posso fare?”*

***...vediamo come  
costruire un percorso  
fatto di progressi, senza  
pretendere la perfezione...***

# Soluzione: *Studiare e Costruire!*

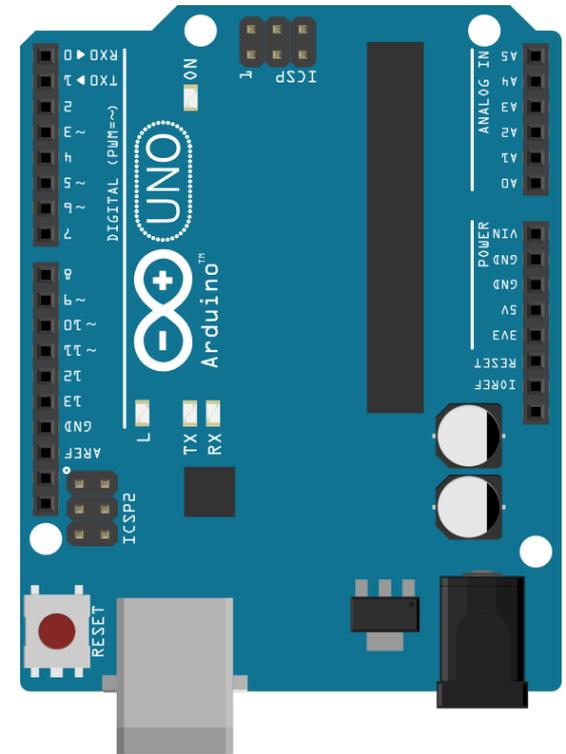
*Le due componenti  
NON sono disgiunte*



# Domanda: ***“Cos’è Arduino?”***

*L'hardware Arduino e il software di programmazione costituisce un sistema di prototipazione elettronica adatto per:*

- *artisti,*
- *designer,*
- *appassionati di elettronica,*
- *hackers,*
- *e tutti coloro che intendono creare oggetti di automazione e controllo che interagiscono con l'ambiente.*



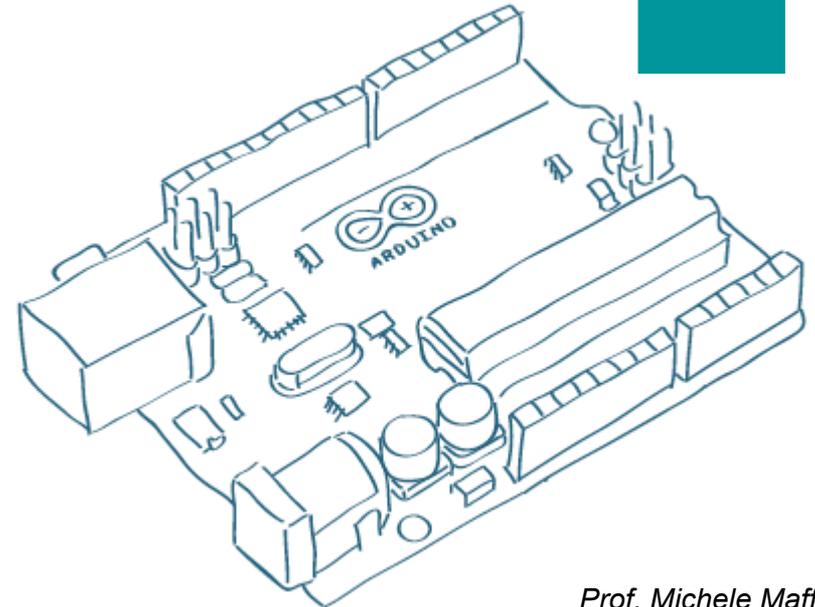
**Domanda:**  
***“perché usare Arduino?”***

Semplicità

1/3

motivi didattici

*...mi consente agevolmente di insegnare un linguaggio di programmazione...*

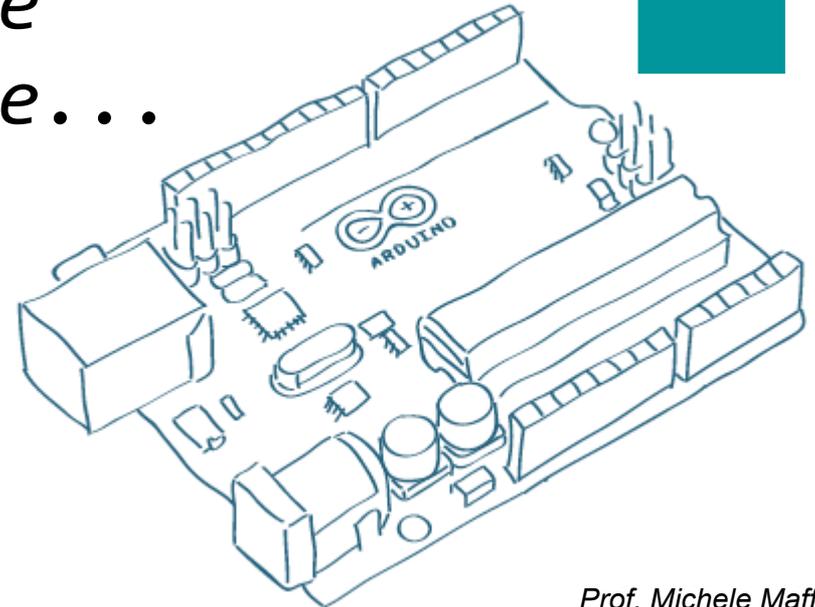


# Semplicità

2/3

motivi didattici

*...mi permette di introdurre agevolmente **concetti di base di elettronica** anche con allievi che non hanno competenze matematiche approfondite...*

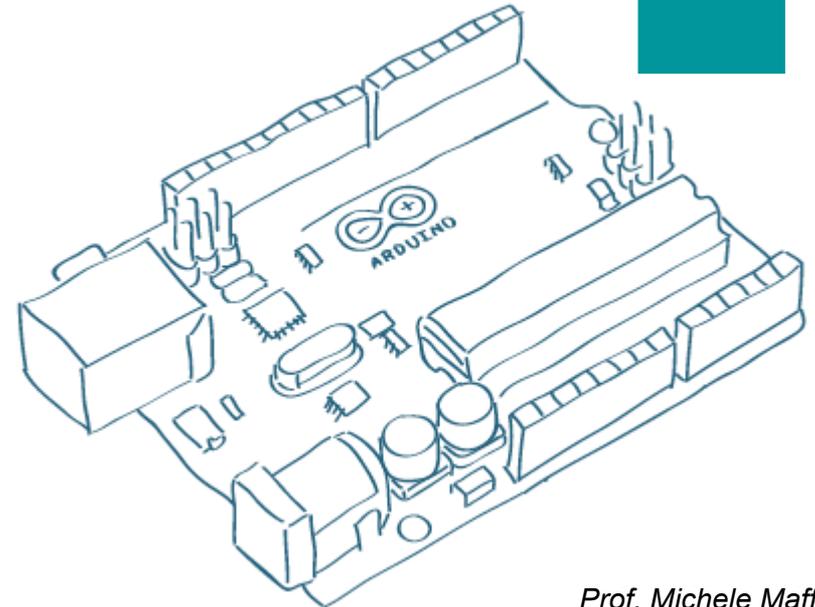


# Semplicità

3/3

motivi didattici

*...si possono realizzare in breve tempo dispositivi che provocano soddisfazione nell'allievo...*

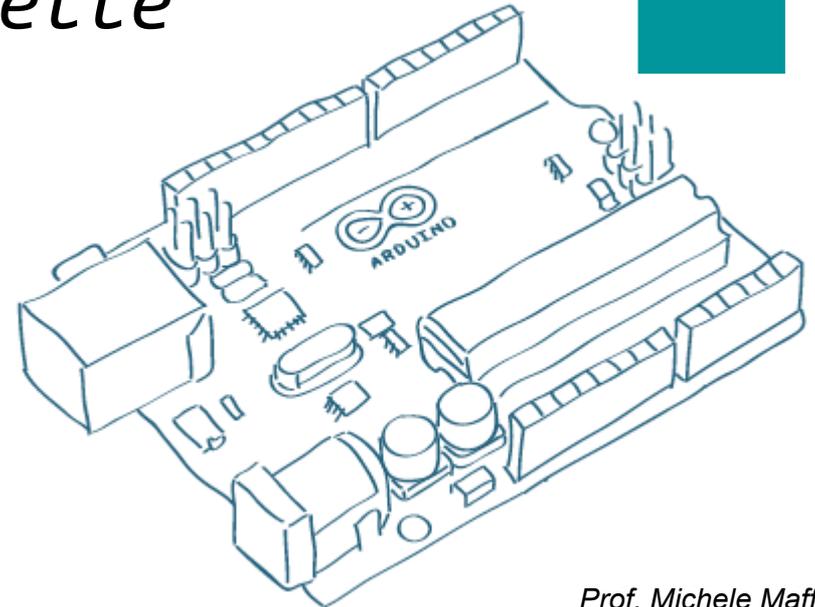


# Documentazione

1/2

motivi didattici

*...disponibilità di moltissima documentazione e la sua grande diffusione permette di soddisfare gran parte delle esigenze...*

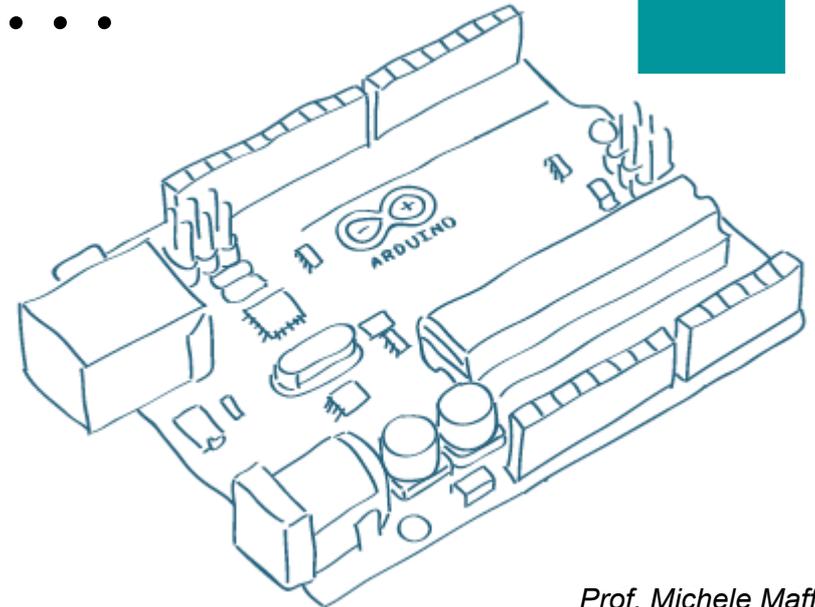


# Documentazione

2/2

motivi didattici

*...ciò provoca voglia di sperimentazione e ricerca favorendo la creazione di artefatti più complessi...*

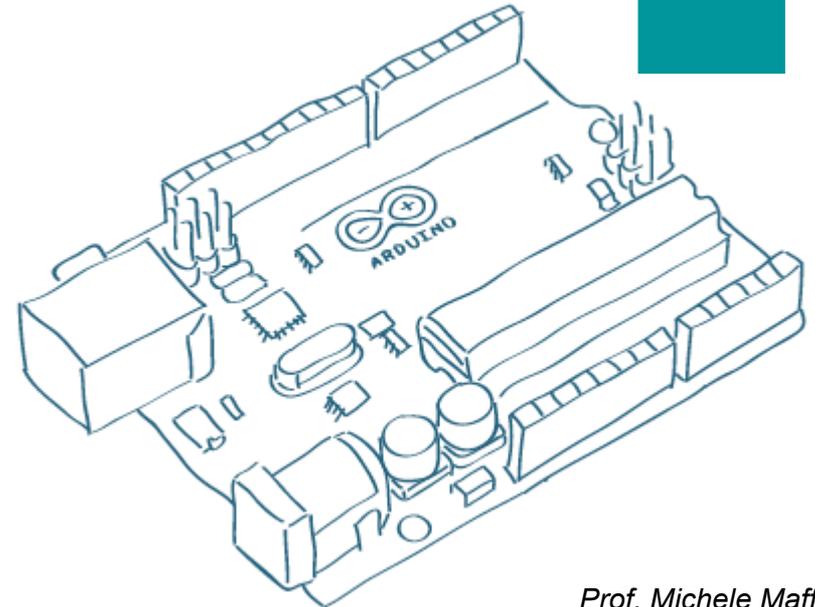


# Motivazione

1/2

motivi didattici

*...stimolo motivazionale  
per lo studio di altre  
discipline (matematica, fisica,  
meccanica, ecc...)*

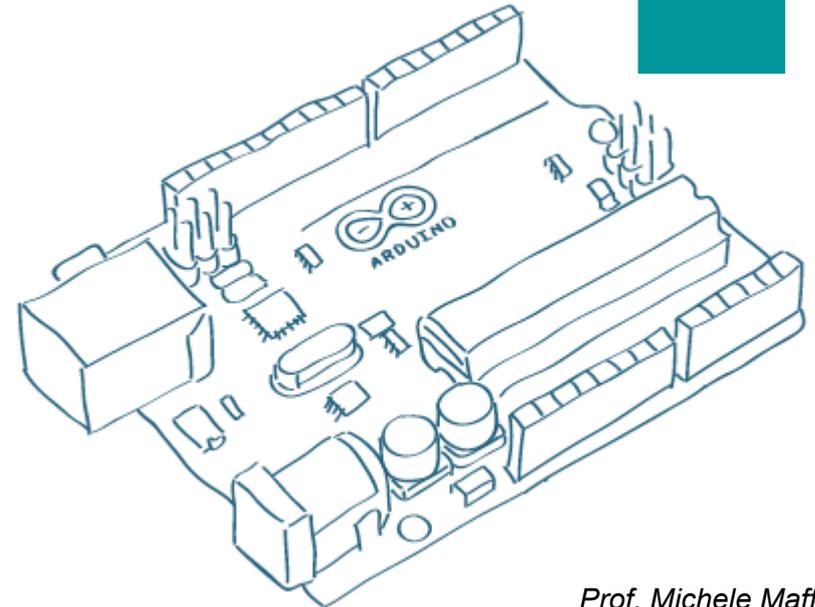


# Motivazione

2/2

motivi didattici

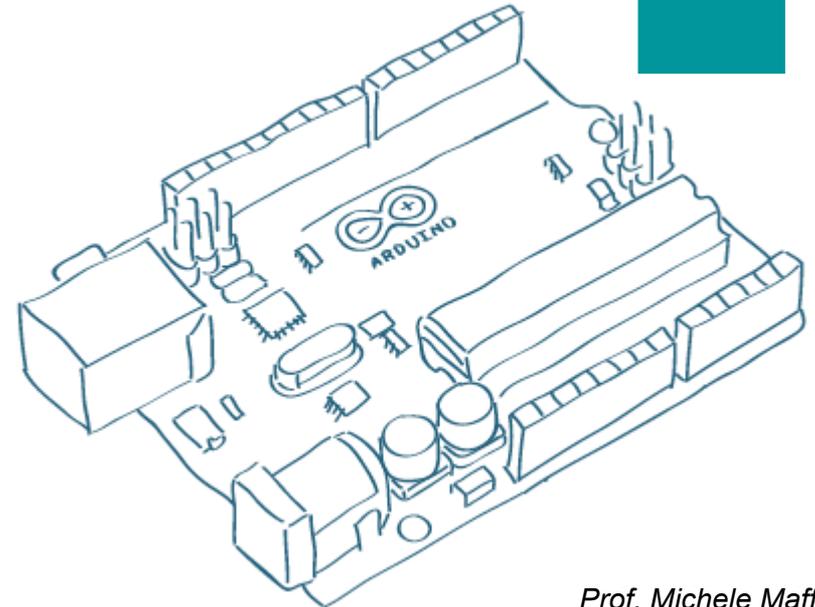
*...quindi Arduino utilizzato  
per **imparare ad imparare***



# Costi

motivi di progetto

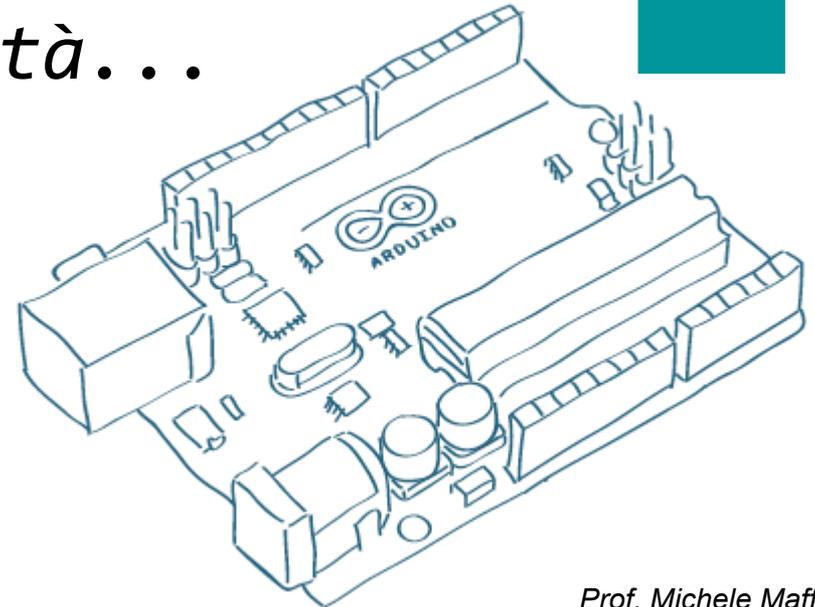
*...costi estremamente bassi  
(poche decine di Euro) per  
produrre sistemi di automazione  
complessi...*



# Espansione

motivi di progetto

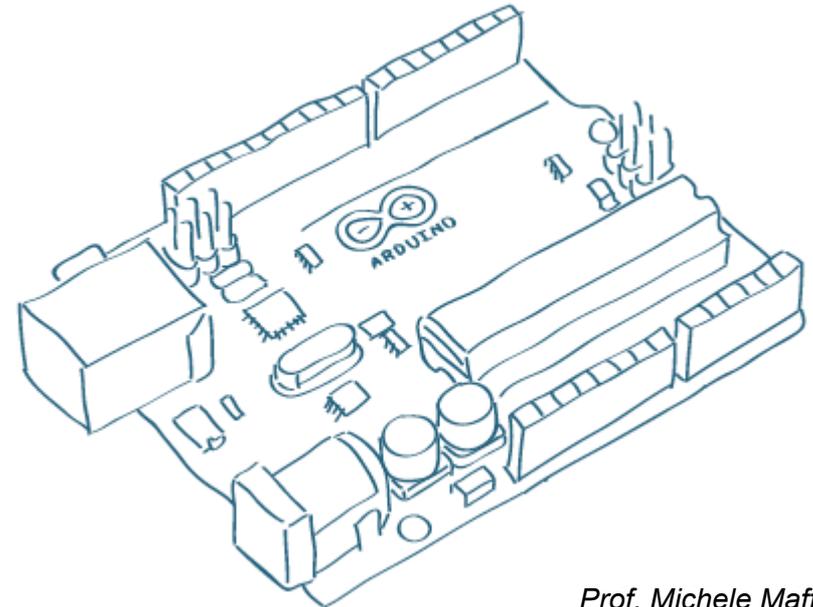
*...disponibilità di un gran numero di schede (shield), anche a basso costo che ne estendono le funzionalità...*



# Prototipazione

motivi di progetto

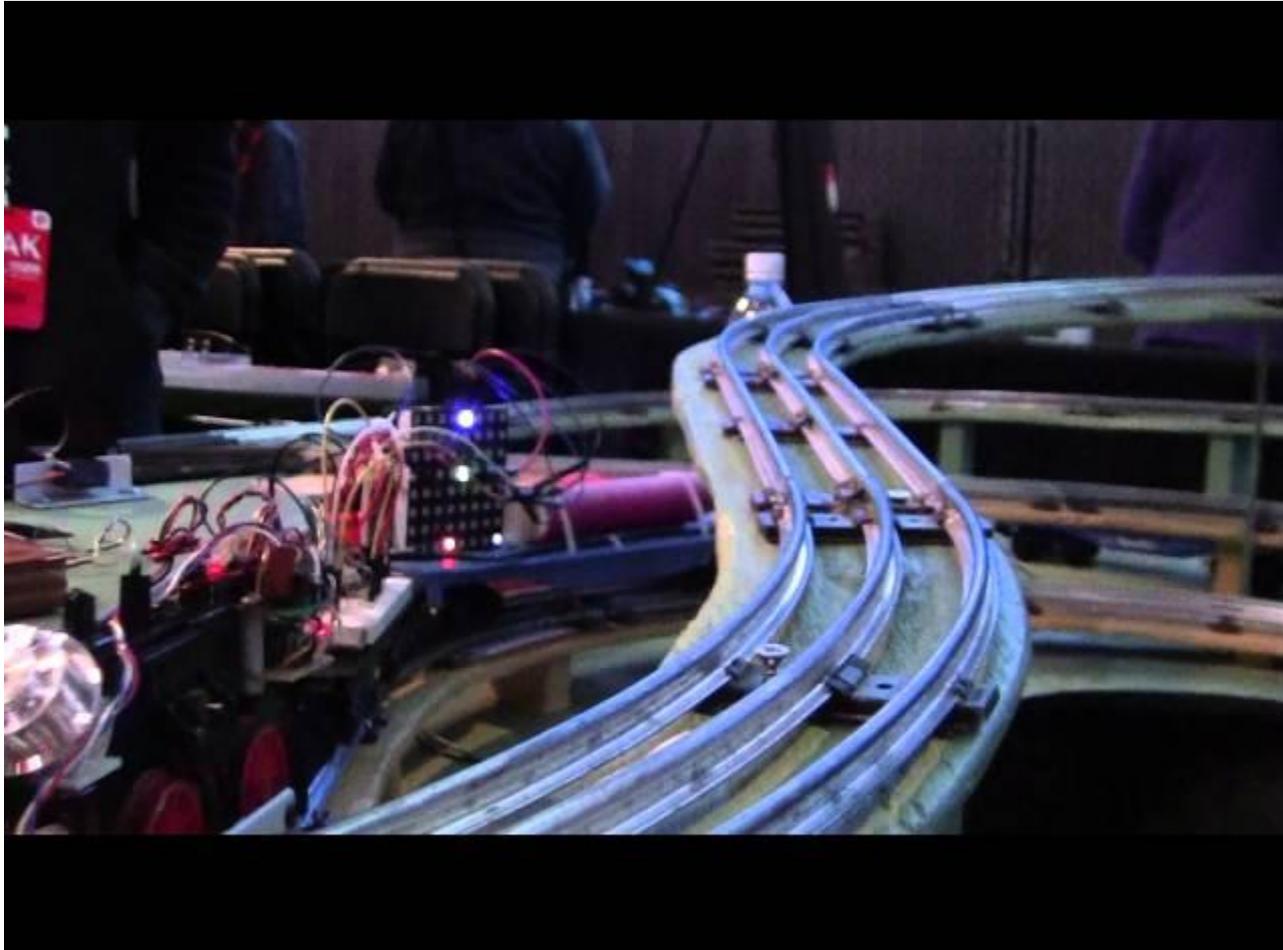
*...rapidità di prototipazione  
di circuiti elettronici che  
possono poi andare in  
produzione...*



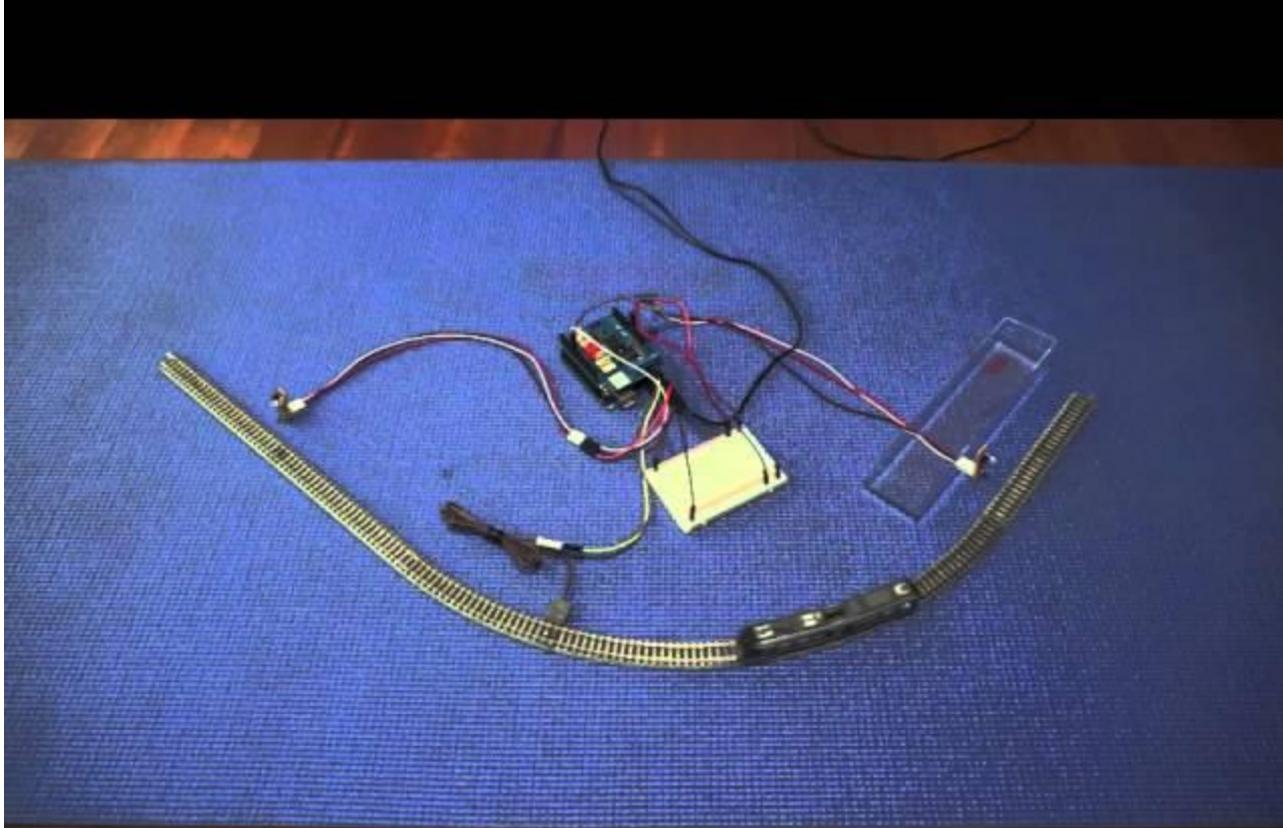
# **Cosa posso fare con Arduino**

**alcune idee per il modellismo**

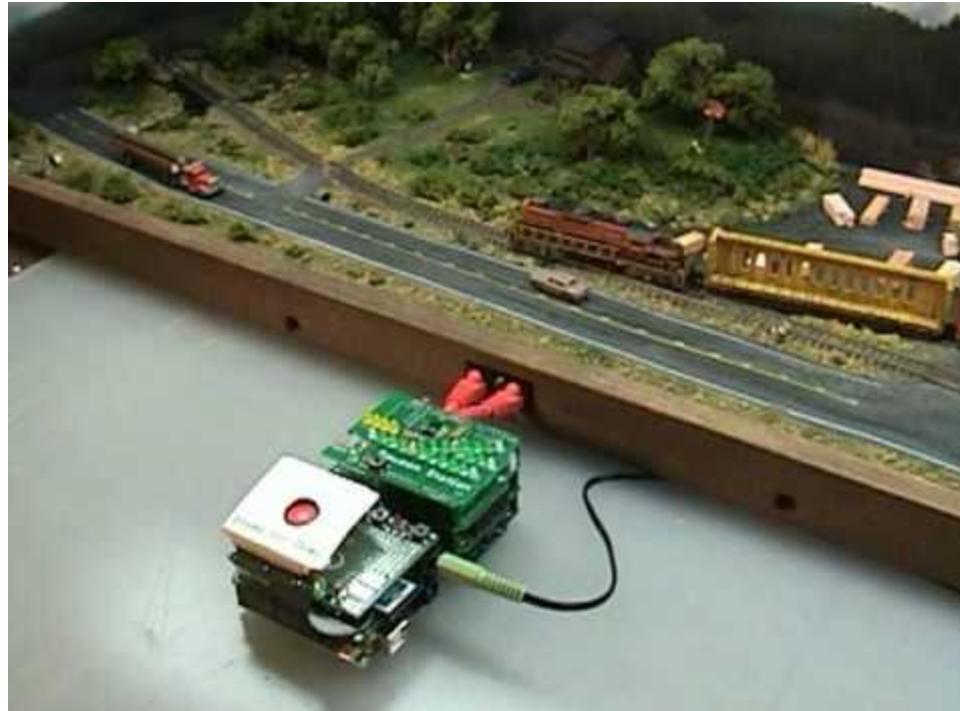
# Controllo velocità treni con scheda Arduino su locomotore



## Controllo direzione treni con sensori ad infrarossi



Controllo di un plastico con treno e pilotaggio treno con Wii Nunchuk



## Automobili radiocomandate



## Veicoli pilotati remotamente via WiFi con telecamera



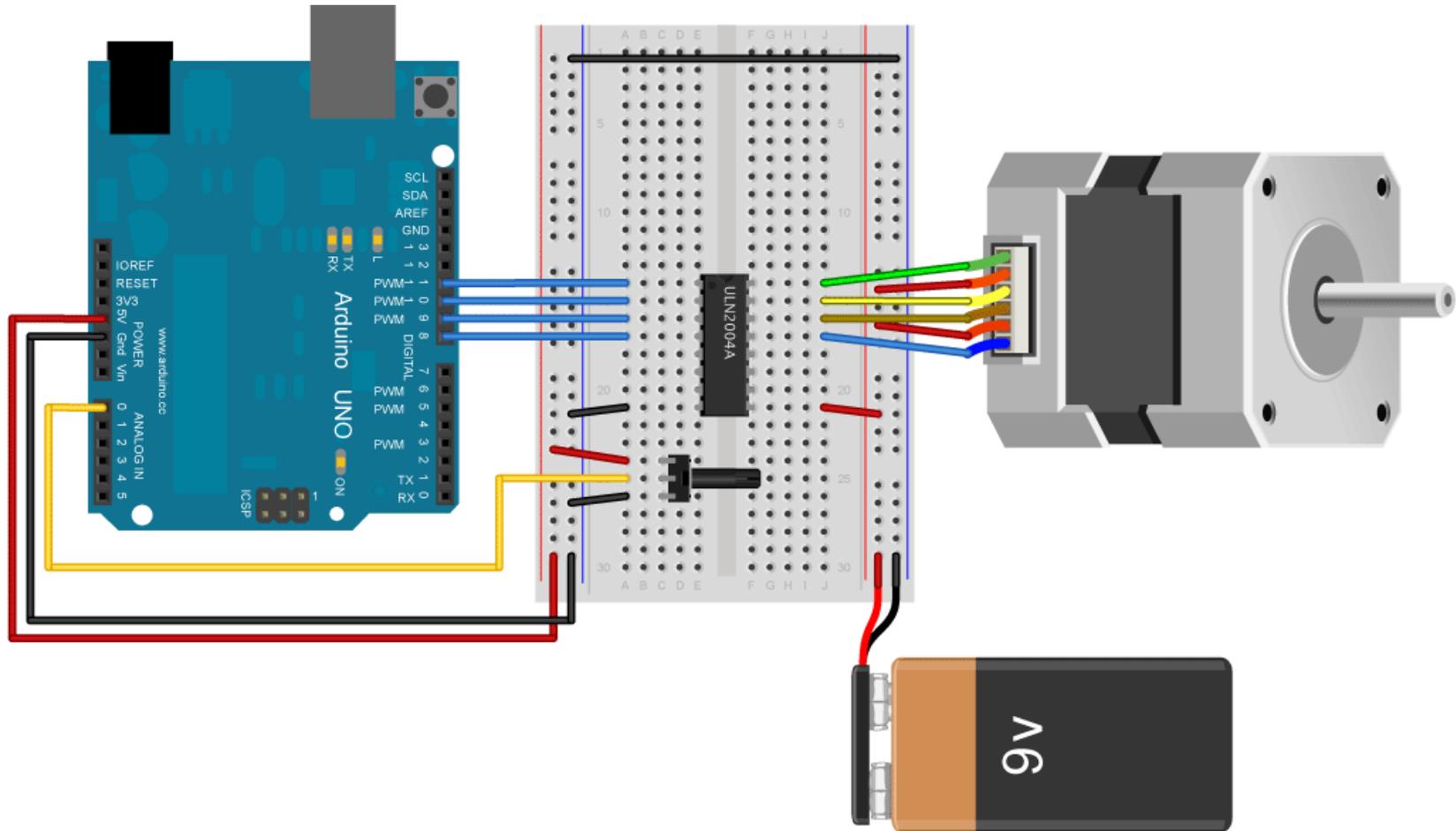
OWI 535 Robotic Arm Edge, Turned into Rover  
Arm Robot. Arduino



Arduino Hexapod spider robot kit with servo control board

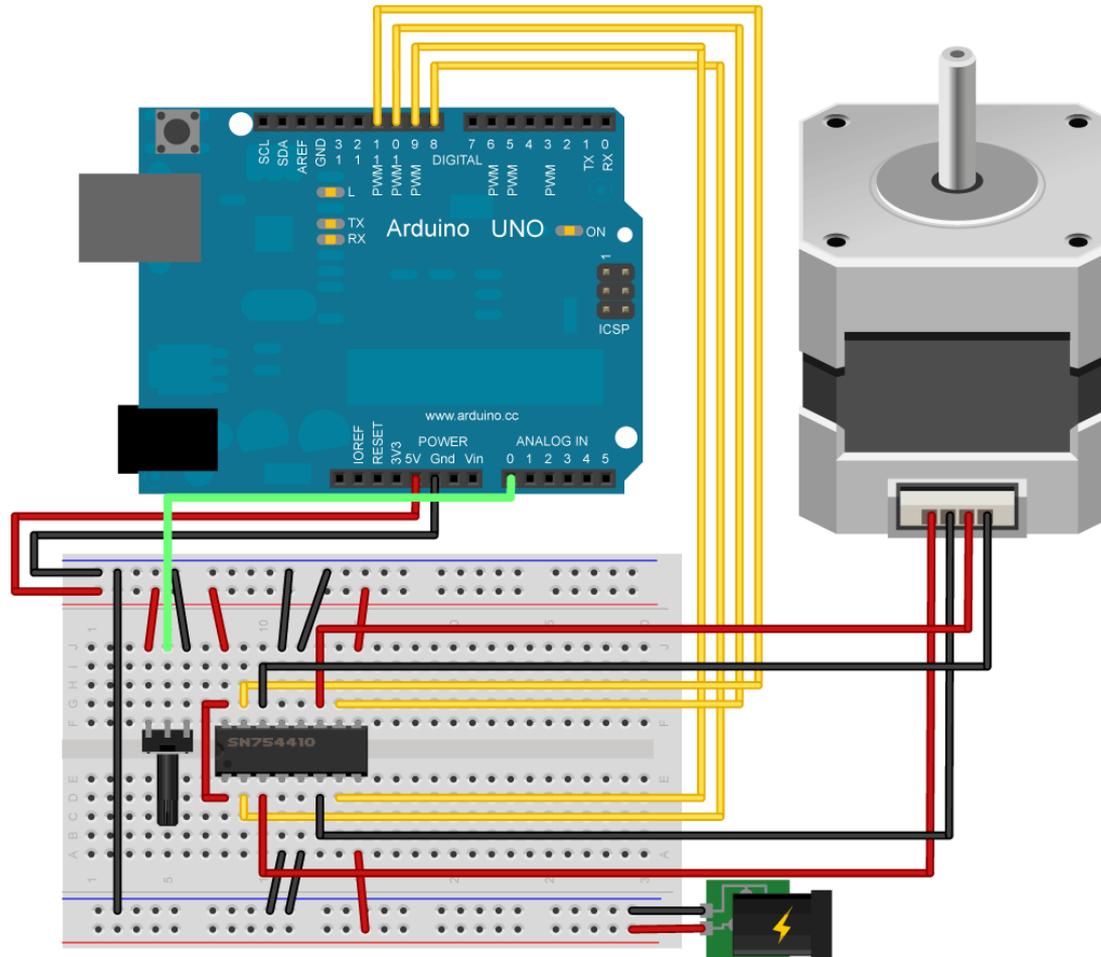


Pilotaggio di motori stepper unipolari



Made with  Fritzing.org

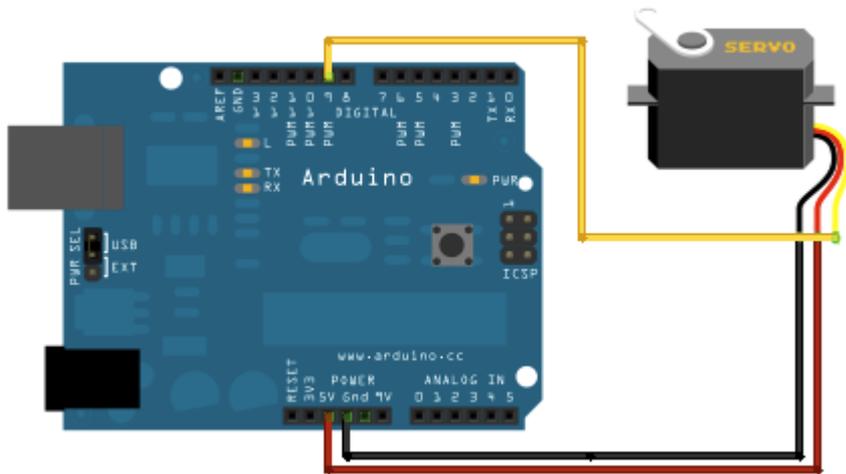
Pilotaggio di motori stepper bipolari



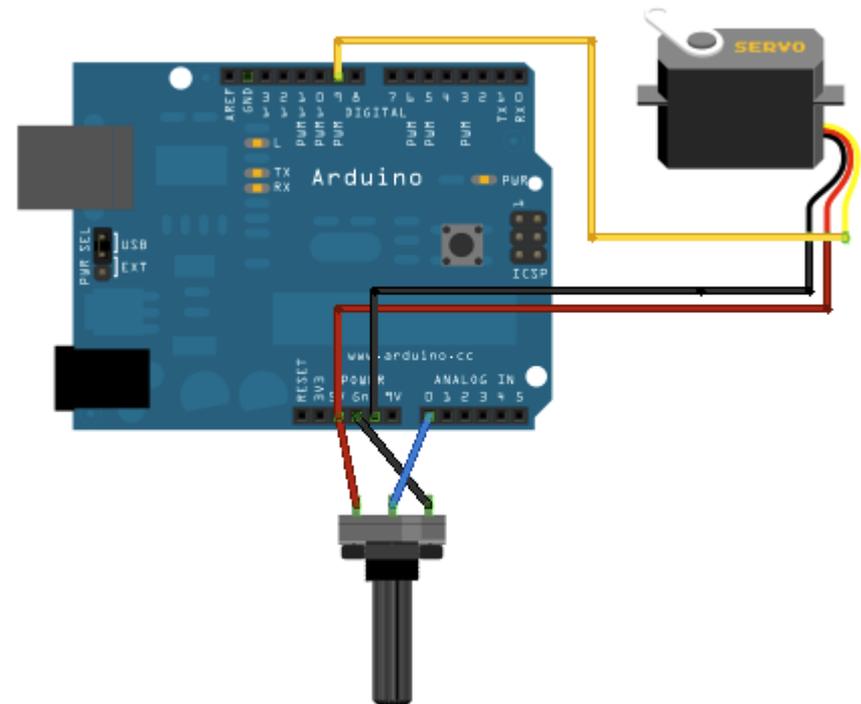
Made with  Fritzing.org

## Pilotaggio di servomotori

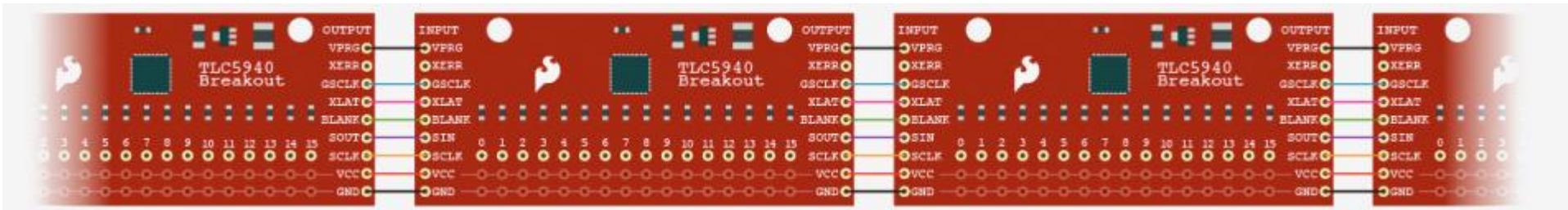
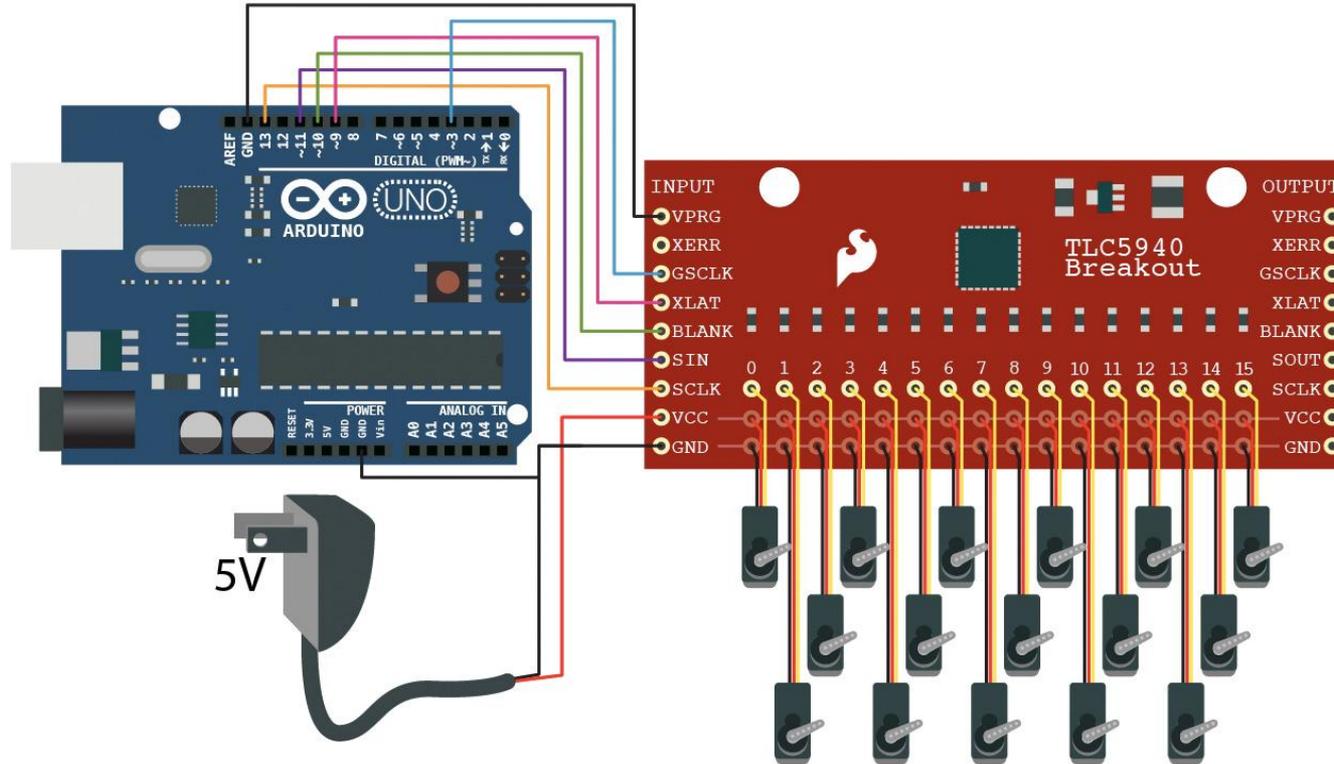
<http://arduino.cc/en/Tutorial/Sweep>



<http://arduino.cc/en/Tutorial/Knob#>



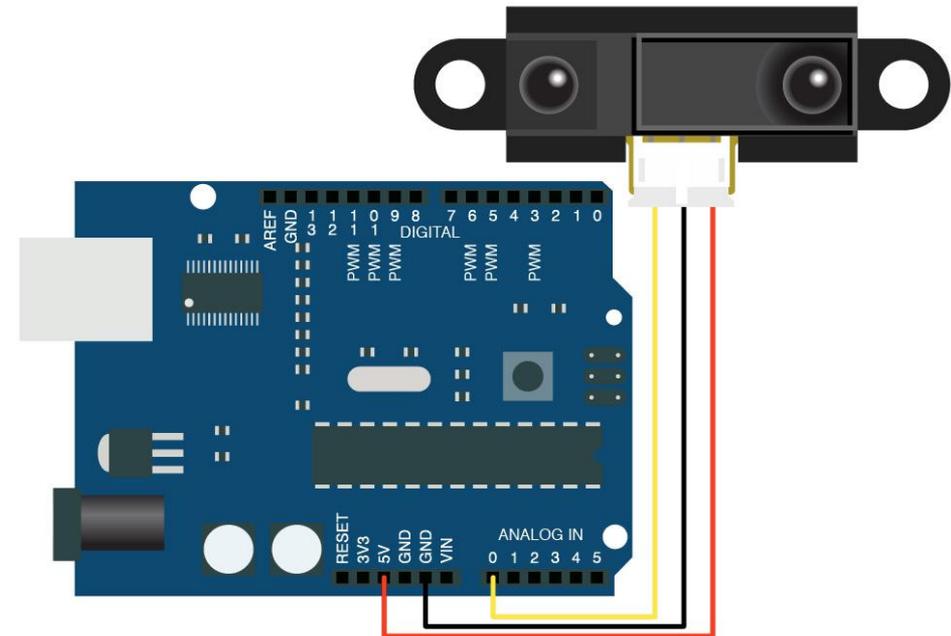
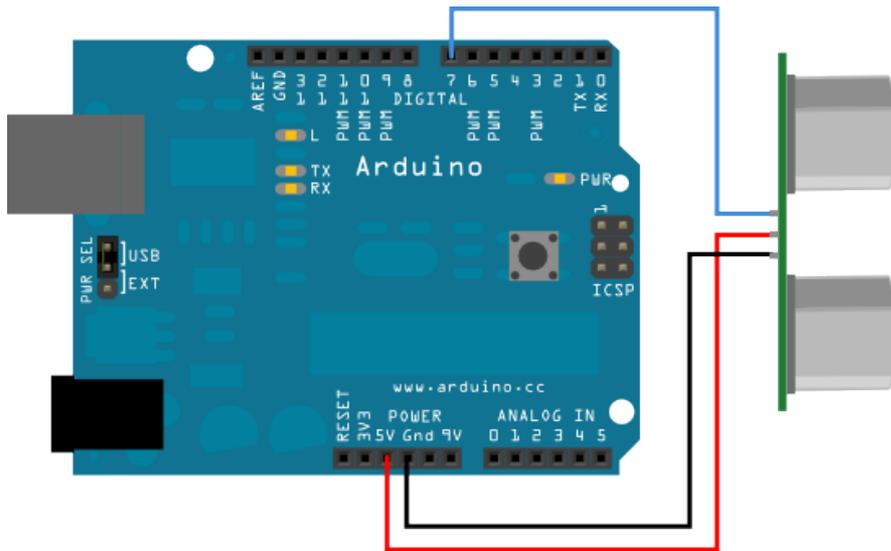
Controllo contemporaneo di una grande quantità di servomotori



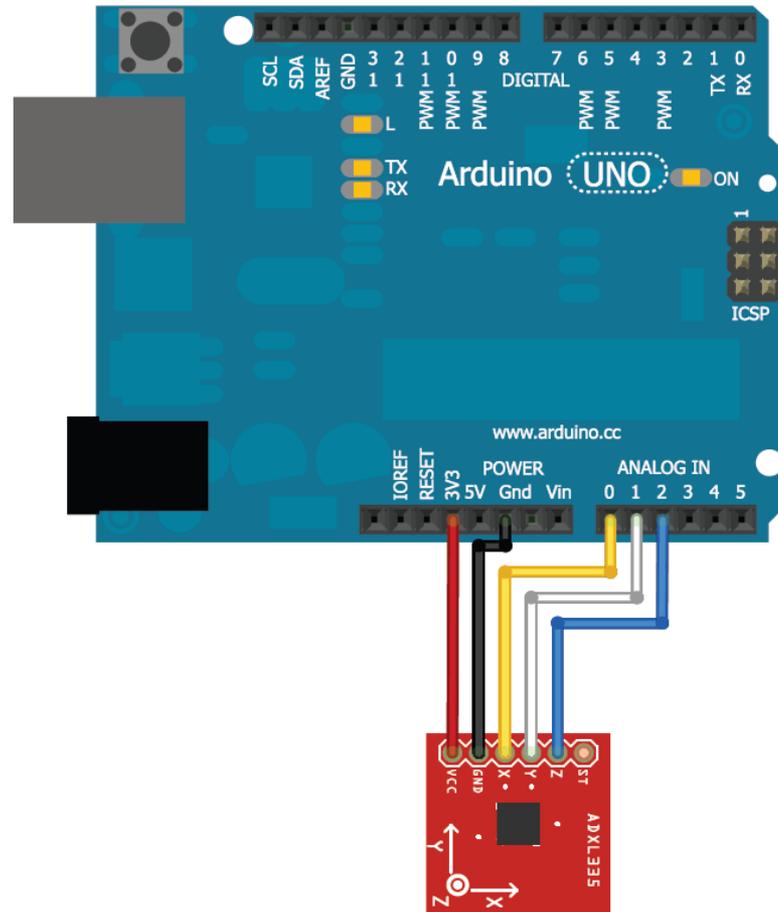
Sensori ad ultrasuoni e ad infrarossi

Sensore ad ultrasuoni

Sensore ad ad infrarossi

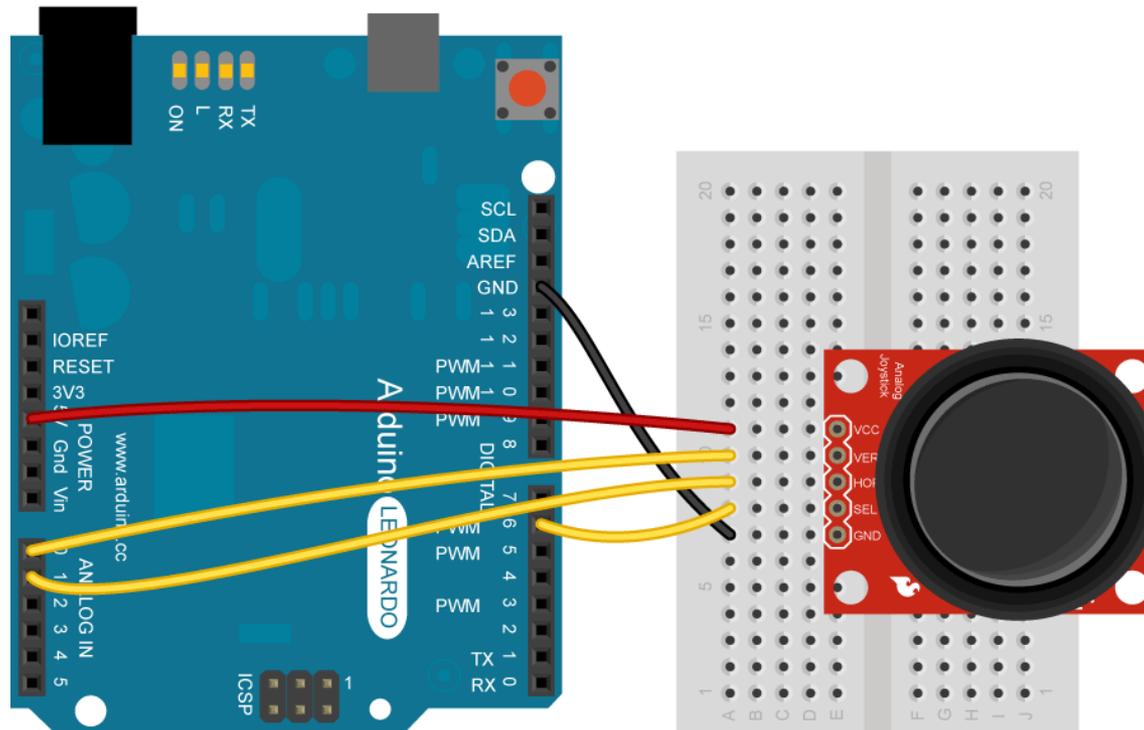


# Accelerometro

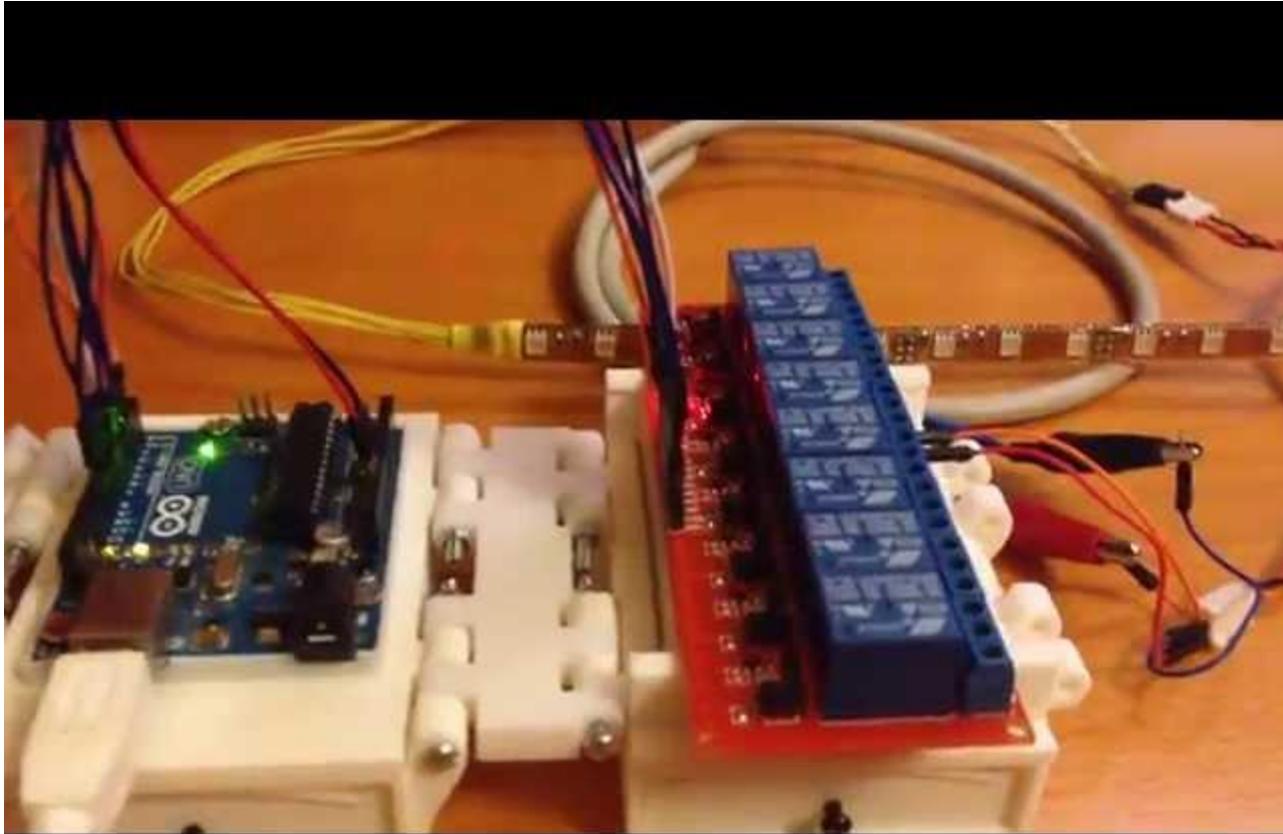


## Joystick Mouse Control

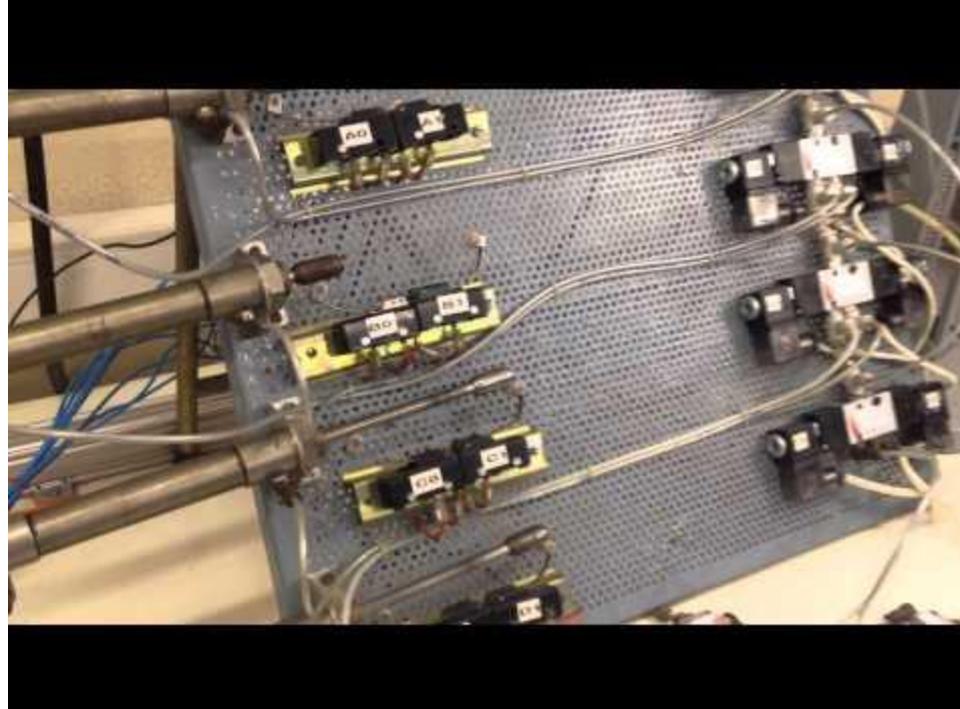
Per il controllo del cursore del computer o il pilotaggio di altro dispositivo esterno



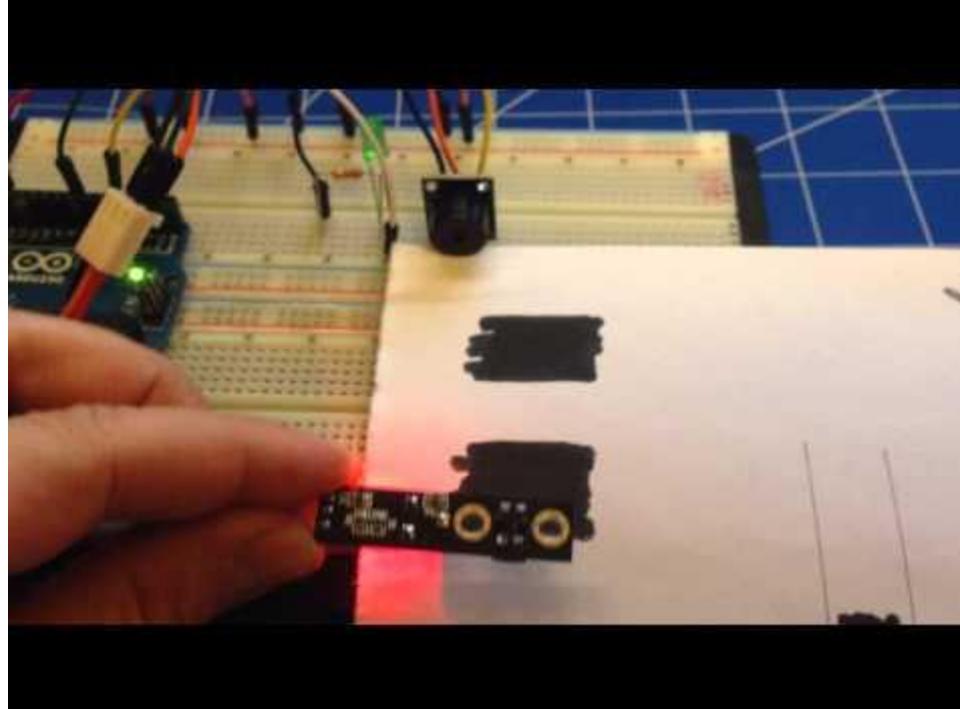
## Controllo relè



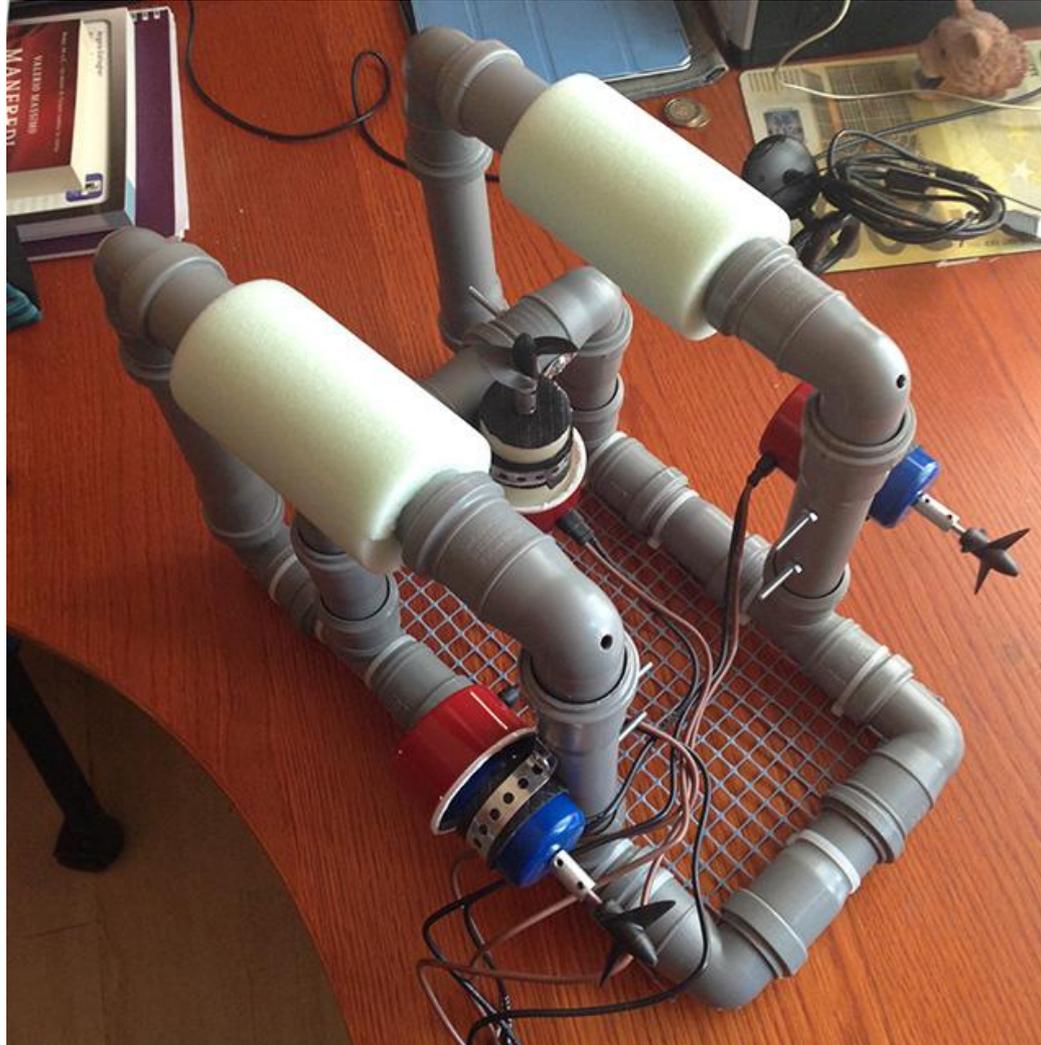
## Controllo di sistemi pneumatici



## Sensor tracking



ROV (Remotely Operated Underwater Vehicle)



*...e molto altro*

*La grandissima quantità di schede elettroniche disponibili sul mercato e la semplicità nel gestirle mediante Arduino rende più semplice costruire dispositivi in grado di soddisfare un gran numero di esigenze di automazione...*

*...ma per superare le difficoltà derivanti da competenze non elevate in elettronica e programmazione è indispensabile studiare e sperimentare.*

**Avviso  
per non spaventarsi  
:-)**

*Imparare ad usare Arduino è cosa semplice i tutorial che trovate on-line vi guideranno ad una comprensione dei concetti di base che potranno soddisfare molte vostre esigenze, ma...*

*per chi vuole approfondire alcune indicazioni...*

# **Competenze di base**

**Dove imparo  
l'elettrotecnica  
e l'elettronica?**

# Siti

# All About Circuits



## Latest Forum Posts

[Paralleling PSUs](#)  
[On/Off switch on electric sewing machine](#)  
[Transformer pinout /connection](#)  
[555 voltage output low](#)  
[Solar Light: supplementing charging to battery](#)



## Welcome to All About Circuits

This site provides a series of online textbooks covering electricity and electronics. The information provided is great for both students and hobbyists who are looking to expand their knowledge in this field. These textbooks were written by Tony R. Kuphaldt and released under the [Design Science License](#). Interested in contributing to the textbooks? Please [click here](#).

### Textbook - Lessons in Electric Circuits

#### Direct Current

> Basic Concepts of Electricity, OHM's Law, Electrical Safety, Scientific Notation...

#### Digital

> Binary Arithmetic, Logic Gates, Switches, Boolean Algebra, Karnaugh Mapping...

#### Alternating Current

> Basic AC Theory, Complex Numbers, Reactance and Impedance, Resonance...

#### References

> Conversion Factors, Resistor Color Codes, Math References, Periodic Table...

#### Semiconductors

> Amplifiers and Active Devices, Solid-State Device Theory, Diodes and Rectifiers...

#### Experiments

> Test Equipment, DC Circuits, AC Circuits, Discrete Semiconductor Circuits...

### Forums

Our [Electronics Forum](#) is a place where hundreds of thousands of students, hobbyists and professionals from around the world share knowledge and ideas. We encourage you to [sign-up](#) and become part of our community!

### Videos

Thanks to a contribution to AAC from Tim Fiegenbaum, we have over [100 Video Lectures](#) based on the textbook "Electronics for Computer Technology". These video lectures cover many of the topics contained within the textbooks, and are an excellent supplement to reinforce concepts.

### Worksheets

The [worksheets](#) were created by Tony R. Kuphaldt, and contain hundreds of categorized questions and answers. They provide a great way to test your knowledge and prepare for exams.

# Michele Maffucci - area studenti

CONTATTAMI - RSS

Social network: Twitter - Delicious - Google+ - Facebook - YouTube - Flickr - LinkedIn

## Michele Maffucci

... my stories, life and work



Home Info **Area Studenti** Area riservata studenti Robot Pet Therapy Site map

### Area Studenti

In questa sezione trovi un elenco di documenti **pdf**, **video** e **link a siti** che ti serviranno per lo studio dell'elettronica e dell'elettrotecnica, inoltre potrai accedere al **laboratorio virtuale di elettronica** composto da una raccolta di risorse interattive che ti consentiranno di apprendere in modo semplice e visuale i concetti fondamentali di elettronica.

#### Corso di elettrotecnica ed elettronica



- [Introduzione](#)
- [Lezione 1 – Richiami di Matematica – operazioni con le potenze](#)
- [Lezione 2 – Richiami di Matematica – calcoli numerici](#)
- [Lezione 3 – Richiami di Matematica – divisioni](#)
- [Lezione 4 – Richiami di Matematica – proporzioni](#)
- [Lezione 5 – Richiami di Matematica – circonferenza e cerchio](#)
- [Lezione 6 – Richiami di Matematica – funzioni e diagrammi cartesiani](#)

#### Iscriviti al blog tramite e-mail

Inserisci il tuo indirizzo e-mail per iscriverti a questo blog, e ricevere via e-mail le notifiche di nuovi post.



#### Cerca

#### Categorie

Seleziona una categoria

#### Vuoi aiutare la mia attività di divulgazione?

Se pensi che le informazioni che hai trovato su questo sito siano state utili per il tuo lavoro o per il tuo studio e desideri fare una donazione ti ringrazio.

Perché fare una donazione?

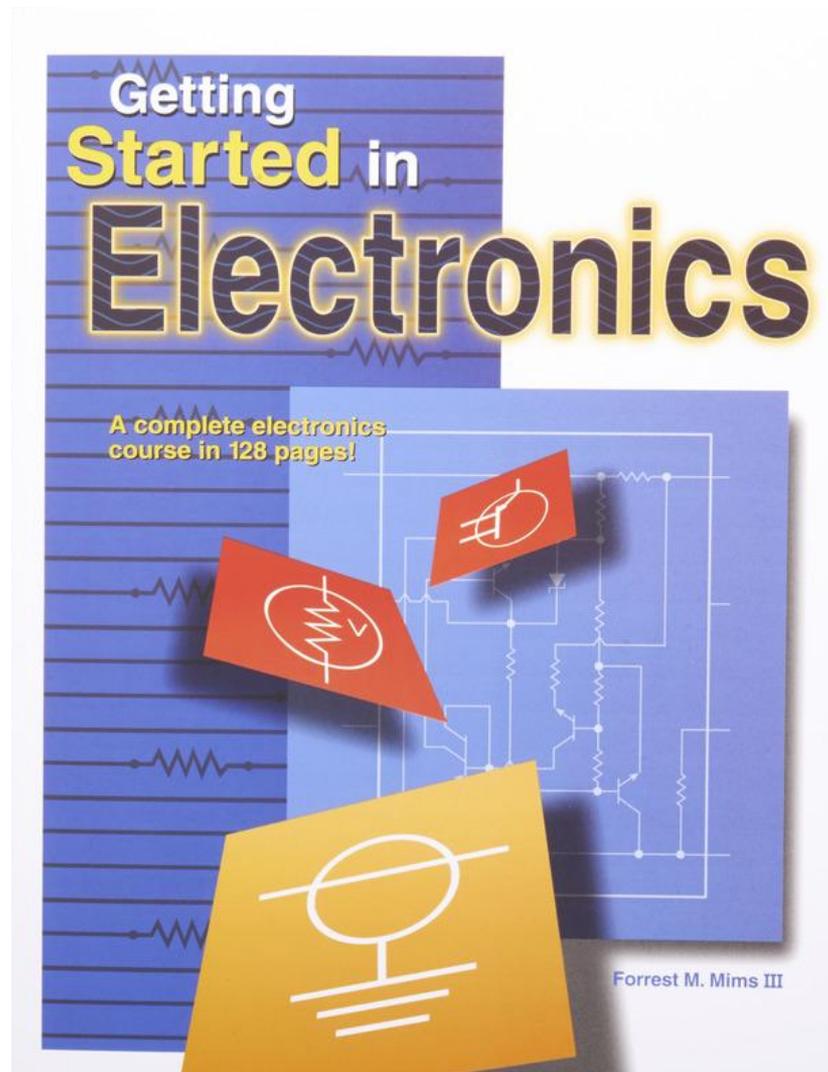
Donazione



# Libri per chi incomincia

essenziali per comprendere velocemente i concetti di base

[Getting Started in Electronics](#)  
di [Forrest M. , III Mims](#)



[Elettronica DIY. La guida per hobbisti e maker - di Pier Calderan](#)





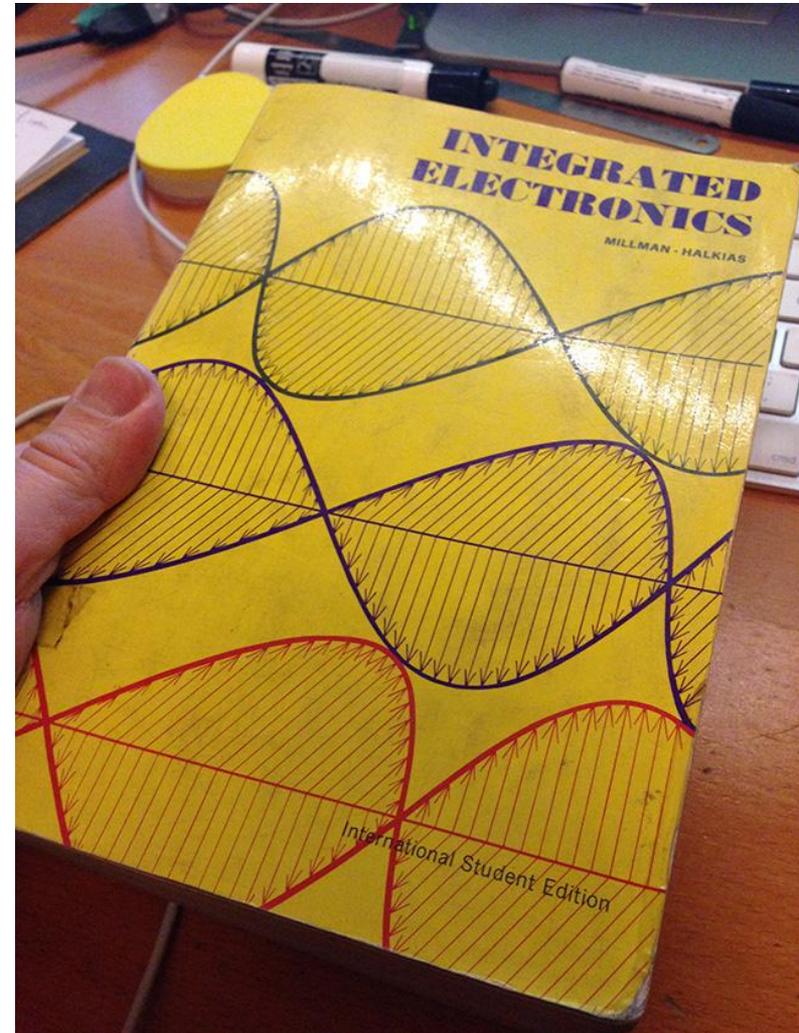
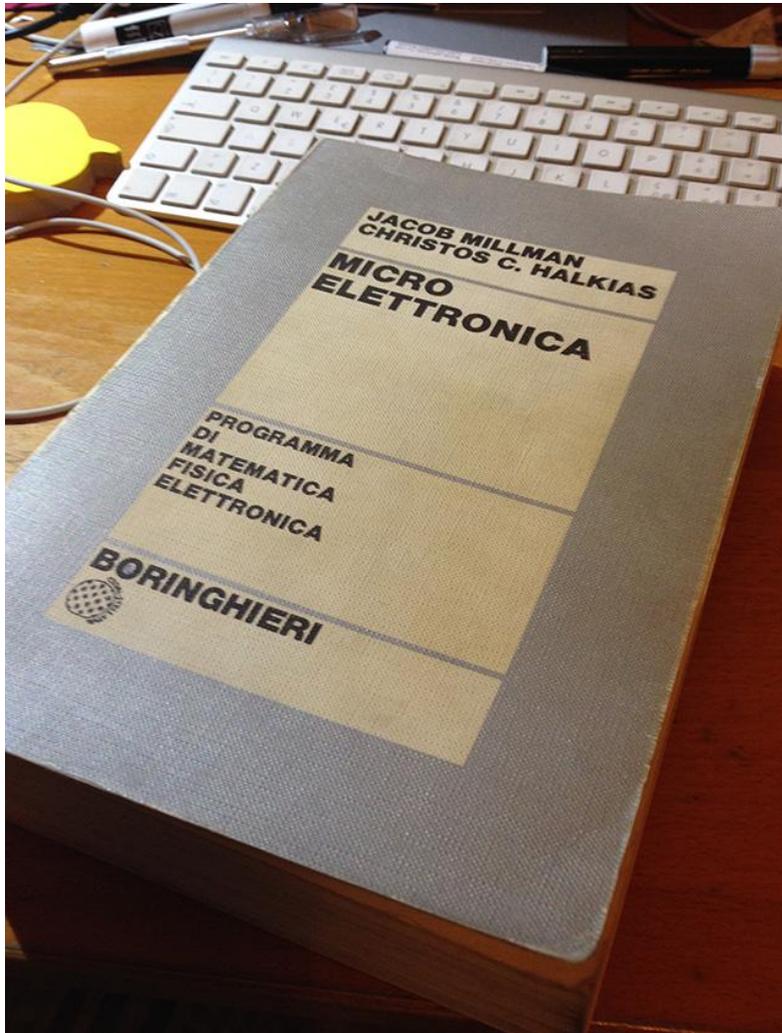
# Libri per approfondire

per assimilare solide basi di elettrotecnica ed elettronica - libri a livello scuola superiore/universitario

P.S.

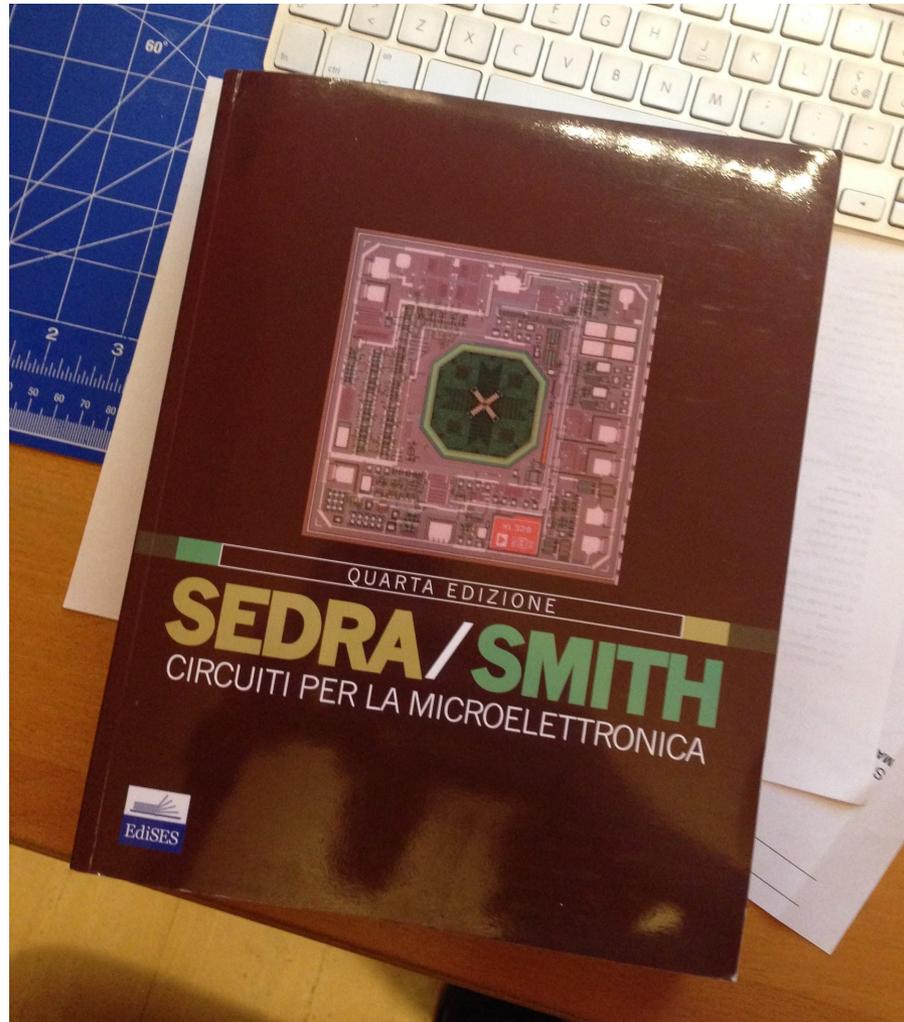
*non è obbligatorio leggere i Libri che seguono per realizzare i tuoi progetti di modellismo con Arduino, ma se vuoi spingerti oltre e raggiungere "l'illuminazione" allora considera questi libri (li trovi sicuramente anche in biblioteca).*

[Micro elettronica di Javob Millman e Christos C. Halkias edito da Boringhieri](#)

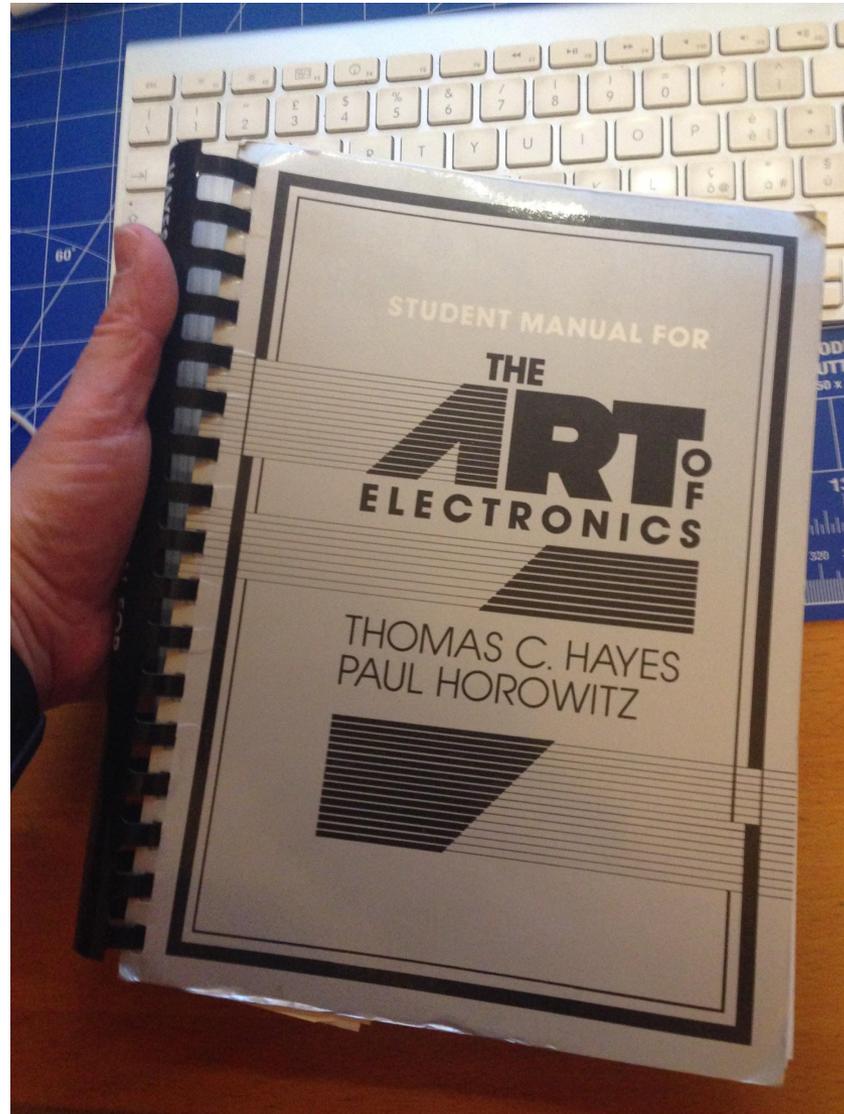


*La "bibbia" dell'elettronica, su di esso si giura eterna fedeltà ai componenti elettronici.*

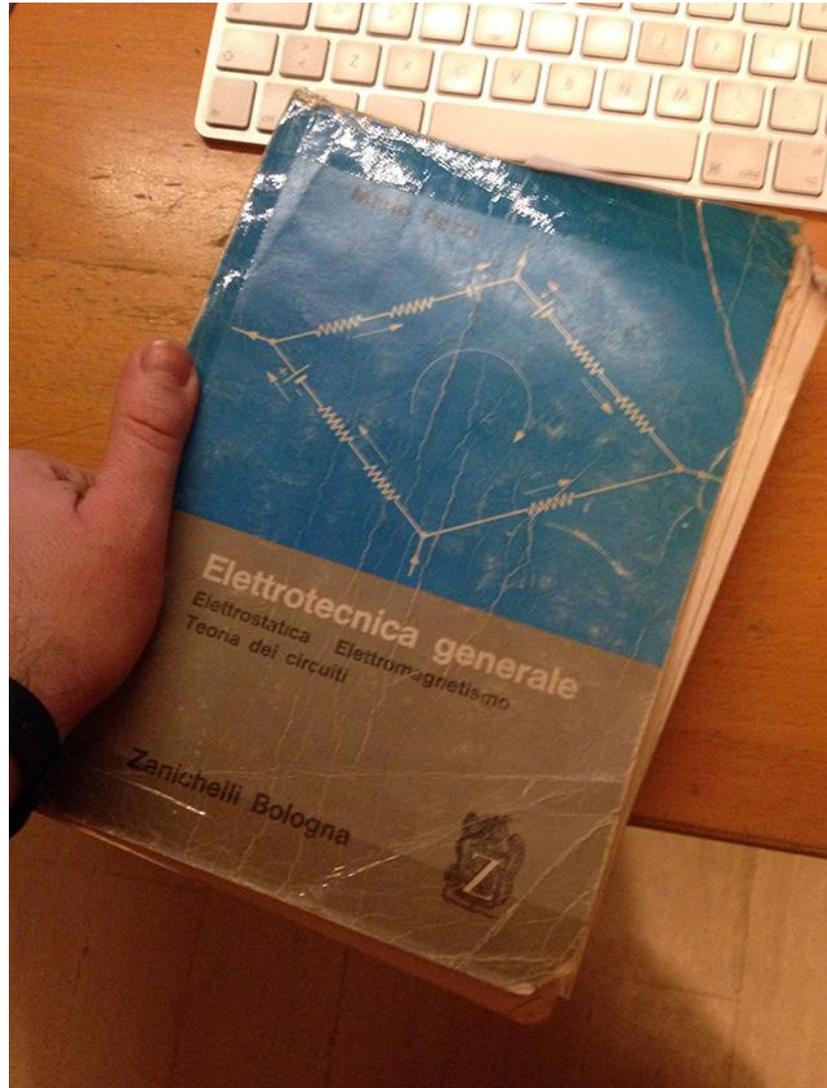
[Circuiti per la microelettronica](#)  
[di Adel S. Sedra, Kenneth C. Smith](#)



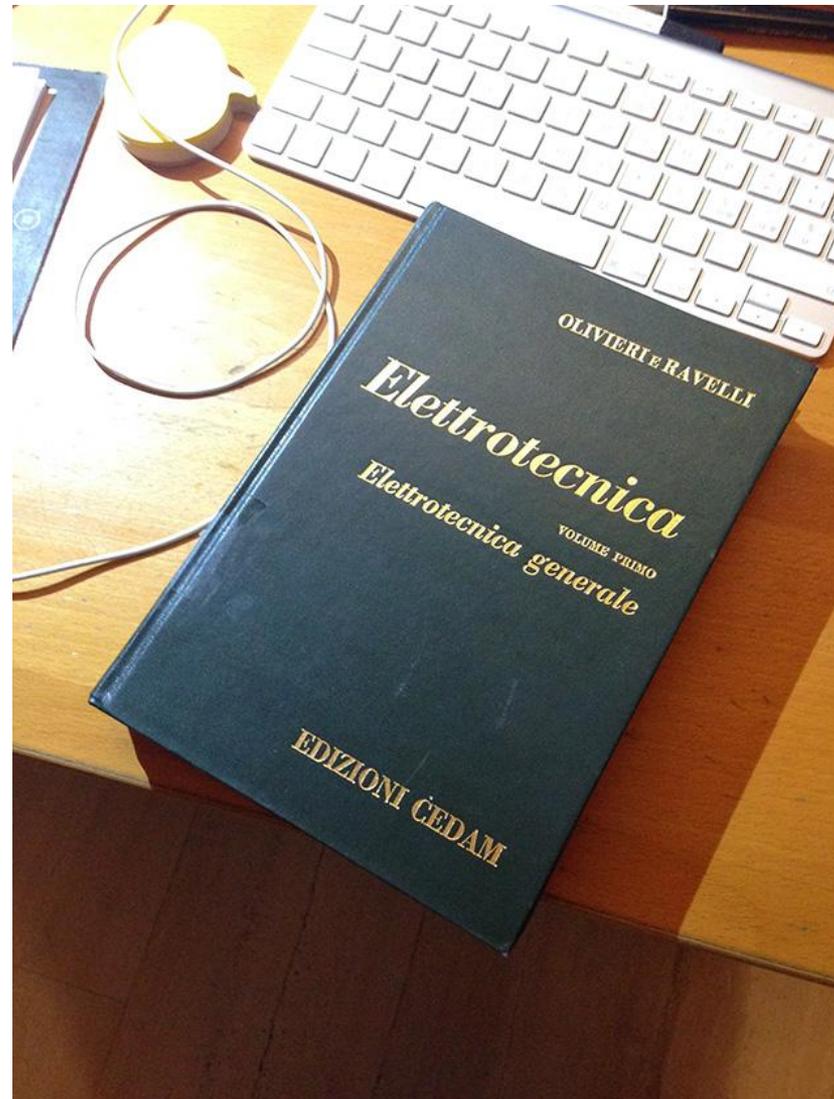
The Art of Electronics Student Manual  
di Thomas C. Hayes (Autore), Paul Horowitz



Elettrotecnica generale - Elettrostatica  
Elettromagnetismo Teoria dei circuiti di Mario  
Pezzi edito da Zanichelli.



Elettrotecnica - Elettrotecnica generale di  
Olivieri e Ravelli edito da Edizioni Cedam.



# Competenze di base

## Dove imparo a programmare

P.S.

*come per l'elettronica anche per imparare a programmare Arduino non è indispensabile essere degli informatici è conoscere già un linguaggio di programmazione, ma se vuoi approfondire considera i consigli che seguono.*

# Libri - esercizi - videocorsi

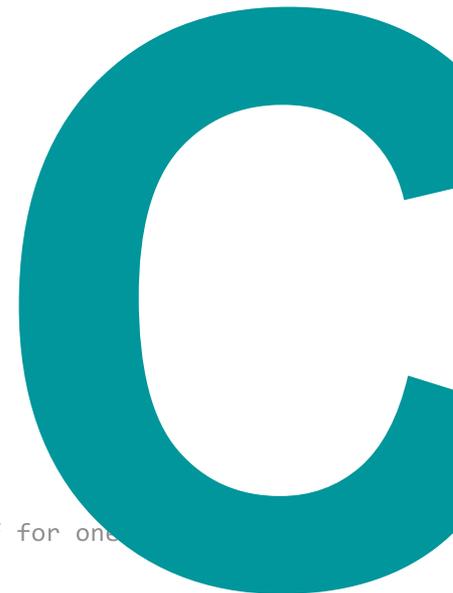
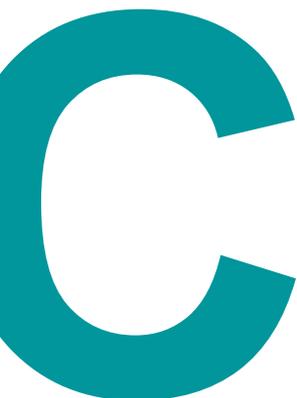
*tutto gratuito*

***...alla base C e C++***

Suggerimenti tratti da articoli scritti su maffucci.it

- [Risorse web per imparare a programmare in C](#)
- [Programmazione in C – slide ed esercizi](#)
- [Manuali gratuiti on-line in italiano di programmazione in C](#)
- [Impariamo il C per programmare i robot](#)
- [Programmazione C++ e dintorni](#)

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second.  
  
  This example code is in the public domain.  
  */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage  
  level)  
  delay(1000); // wait for a second  
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```



## Riferimenti importanti online

- **Prof. Fulvio Corno**
  - [slide del corso di informatica](#)
  - [esercizi di programmazione in C](#)
  - [videocorso linguaggio C](#)
- [Programmazione e linguaggio C](#)
- [The C Library Reference Guide](#)

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second.  
  This example code is in the public domain.  
  */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage  
  level)  
  delay(1000); // wait for a second  
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

# **Competenze di base**

## **Dove imparo l'Alfabeto di Arduino**

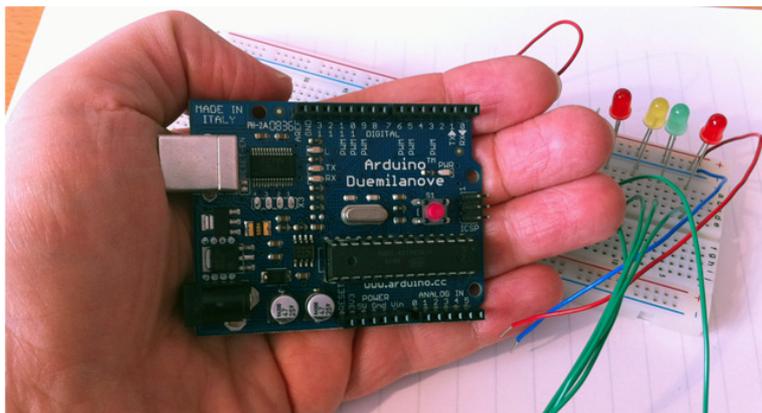
# Libri

[Michele Maffucci - Area studenti - Arduino](#)**Michele Maffucci**

Home Info Area Studenti Area riservata studenti Robot Pet Therapy Site map

**Arduino**

Ultimo aggiornamento 29.09.2014



In questa pagina sono raccolte tutte le risorse su Arduino, tutorial, esperimenti, lezioni. **Le mie lezioni online su Arduino 2009/Arduino UNO (un corso in costruzione per i miei allievi)**

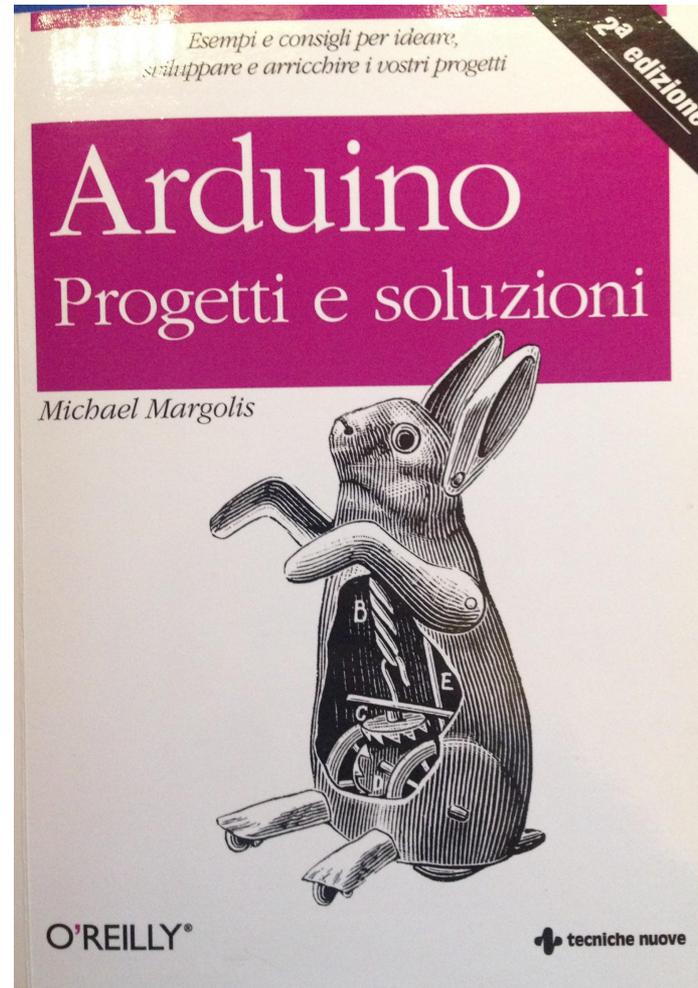
**Libri (free):**

- [Getting Started Section](#)
- [Learning Section](#)
- [Reference Section](#)
- [Arduino Programming Notebook](#)
- [The Complete Beginners Guide to the Arduino](#)
- [Getting started with Arduino: a beginner's guide – by Brad Kendall](#)
- Bionic Arduino:
  - [bionic\\_arduino\\_class1.pdf](#)
  - [bionic\\_arduino\\_class2.pdf](#)
  - [bionic\\_arduino\\_class3.pdf](#)
  - [bionic\\_arduino\\_class4.pdf](#)

**Libri (buy)**

- [Getting Started with Arduino \(Make: Projects\)](#) di Massimo Banzi
- [Arduino e le tecniche di programmazione dei microcontrollori ATMEL \(Edizione Futura Group\)](#) di Michele Menniti
- [Arduino Cookbook](#) di Michael Margolis
- [30 Arduino Projects for the Evil Genius](#) di Simon Monk
- [Practical Arduino: Cool Projects for Open Source Hardware](#) di Jonathan Oxer e Jonathan Oxer
- [Beginning Arduino](#) di Michael McRoberts
- [Building Wireless Sensor Networks: with ZigBee, XBee, Arduino, and Processing](#) di Robert Faludi
- [Programming Interactivity: Unlock the Power of Arduino, Processing, and OpenFrameworks](#) di Joshua Noble
- [Making Things Talk: Practical Methods for Connecting Physical Objects](#) di Tom Igoe
- [Arduino Robotics](#) di John-David Warren, Josh Adams e Harald Molle
- [Make: Arduino Bots and Gadgets: Six Embedded Projects with Open Source Hardware and Software \(Learning by Discovery\)](#) di Tero Karvinen e Kimmo Karvinen
- [Arduino Projects](#) di Instructables Authors
- [Arduino: A Quick-Start Guide \(Pragmatic Programmers\)](#) di Maik Schmidt
- [Cool Arduino Projects: From Home Automation to Rocket Control](#) di Louis Dicarro
- [Practical Arduino: Cool Projects for Open Source Hardware \(Technology in Action\)](#) di Jonathan Oxer e Hugh Blemings
- [Beginning Arduino](#) di Michael McRoberts
- [Arduino: Microcontrollori, Programmazione, Funzionalità, Sintassi, Lezioni on Digital](#)

Arduino. Progetti e soluzioni  
di Michael Margolis



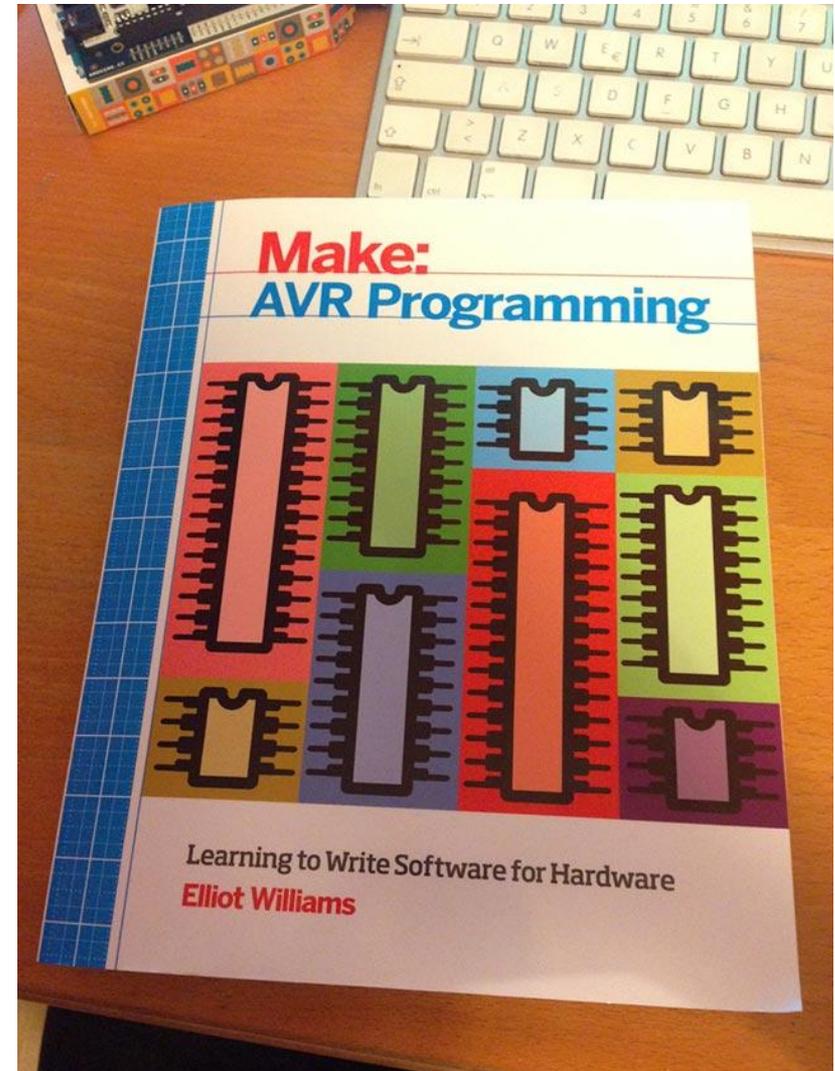
[Il manuale di Arduino di Maik Schmidt](#)



## Make: AVR Programming

Se utilizzate Arduino per i vostri progetti ma avete necessità di maggiori performance e maggior controllo sull'elettronica allora questo potrebbe essere il libro per fare il passo successivo e "tuffarsi" in una programmazione più "spinta" ed imparare le basi sull'uso dei microcontrollori Atmel AVR.

Il libro vi fornirà, mediante un approccio di sperimentazione costante, le basi per apprendere l'uso dei microcontrollori Atmel AVR, scriverete i vostri programmi in C lavorando direttamente sull'hardware eliminando il livello di astrazione offerto dalla programmazione che si potrebbe avere normalmente usando Arduino.



# Corsi on-line

## L'alfabeto Arduino

Corso completo di base sull'uso di Arduino, con sperimentazione e codice sorgente.

GitHub This repository Search

maffucci / **LezioniArduino**

branch: master LezioniArduino / corso01 / +

Inizializzazione variabile temperatura ...

maffucci authored on Mar 27, 2014

- sketch00 Corso Arduino
- sketch01 Corso Arduino
- sketch02 Corso Arduino
- sketch03 Corso Arduino
- sketch04 Corso Arduino
- sketch05a Corso Arduino
- sketch05b Corso Arduino
- sketch05c Corso Arduino
- sketch05d Correzione LED02
- sketch06 Corso Arduino
- sketch06b Uso degli array e gestione del tempo
- sketch07 Corso Arduino
- sketch08 Correzione digitalRead() sul secondo pin

### L'alfabeto di Arduino

Introduzione all'uso di Arduino  
lezione 1  
Prof. Michele Maffucci

1 of 84

### L'alfabeto di Arduino

Introduzione all'uso di Arduino  
lezione 2  
Prof. Michele Maffucci

1 of 102

### L'alfabeto di Arduino

Introduzione all'uso di Arduino  
lezione 3  
Prof. Michele Maffucci

1 of 66

### L'alfabeto di Arduino

Introduzione all'uso di Arduino  
lezione 4  
Prof. Michele Maffucci

1 of 40

[Codice sorgente esercizi](#)

## [Corso on-line e manuali di riferimento](#)

- [Lezione01: Incominciamo con Arduino](#)
- [Arduino – lezione 02: facciamo lampeggiare un led](#)
- [Arduino – lezione 03: controlliamo un led con un pulsante](#)
- [Arduino – lezione 04: realizzare un programma che identifica le variazioni di stato](#)
- [Arduino – lezione 05: controllo presenza](#)
- [Arduino – lezione 06: modulazione di larghezza di impulso \(PWM\)](#)
- [Arduino – lezione 07: lavorare con gruppi di valori e funzioni esterne](#)
- [Arduino – lezione 08: display a 7 segmenti e creazione di librerie](#)
- [Arduino – lezione 09: uso di LED RGB – parte 1](#)

### Appunti di programmazione

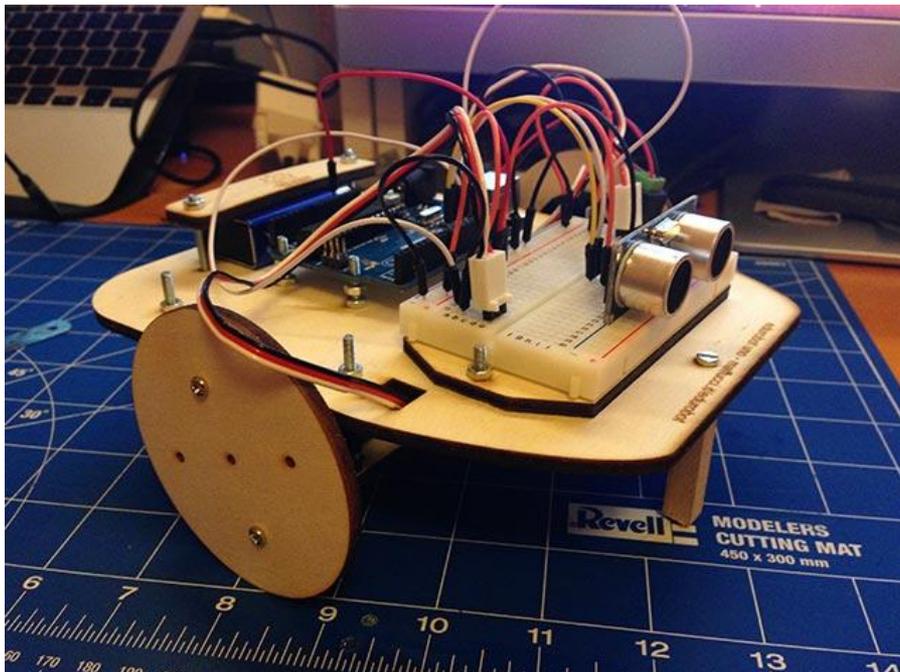


- [strutture](#)
- [variabili](#)
- [tipi di dati](#)
- [aritmetica](#)
- [costanti](#)
- [controllo di flusso](#)
- [ingressi e uscite digitali](#)
- [ingressi e uscite analogiche](#)
- [orologio interno](#)
- [operazioni matematiche](#)
- [numeri casuali](#)
- [comunicazione seriale](#)

## EduRobot - apprendere l'uso di Arduino mediante la robotica

Percorso di formazione alternativo all'uso di Arduino mediante la robotica.

Un percorso che nasce per studenti della scuola superiore volto al miglioramento dell'attenzione e al superamento delle difficoltà che si hanno nelle materie scientifiche.



[Manuale di riferimento - WebApp](#)

Per rendere pratico l'utilizzo del manuale di riferimento è stata realizzata una versione portabile per dispositivi mobili **iOS** e **Android**, maggiori informazioni possono essere trovate seguendo il [link](#).



# Applicativi essenziali

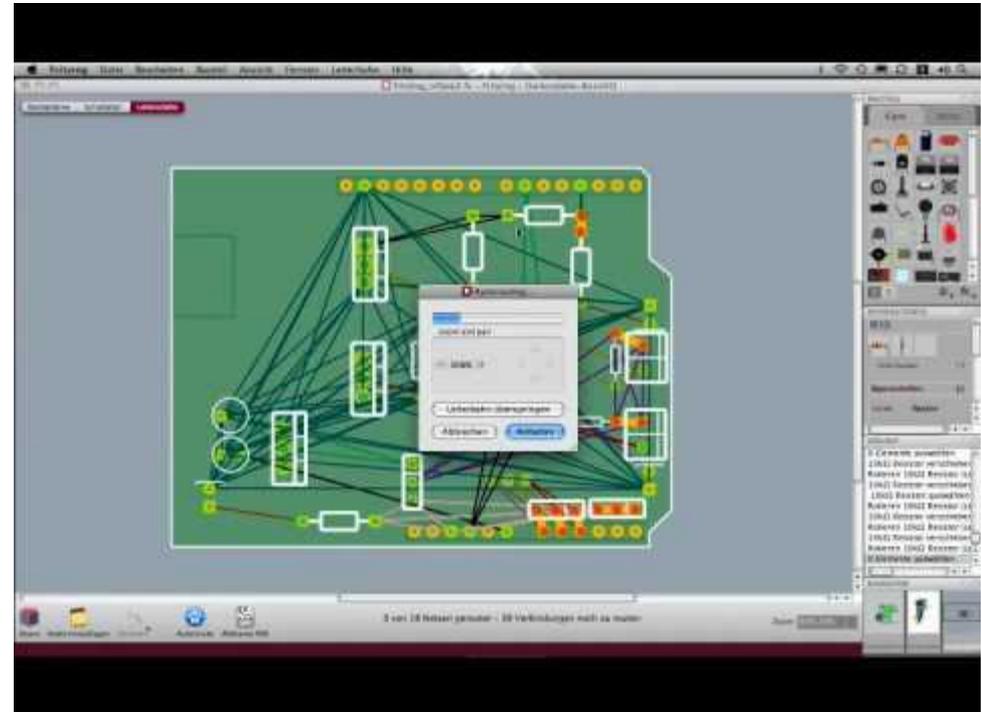
per lo studio e la progettazione

# Fritzing

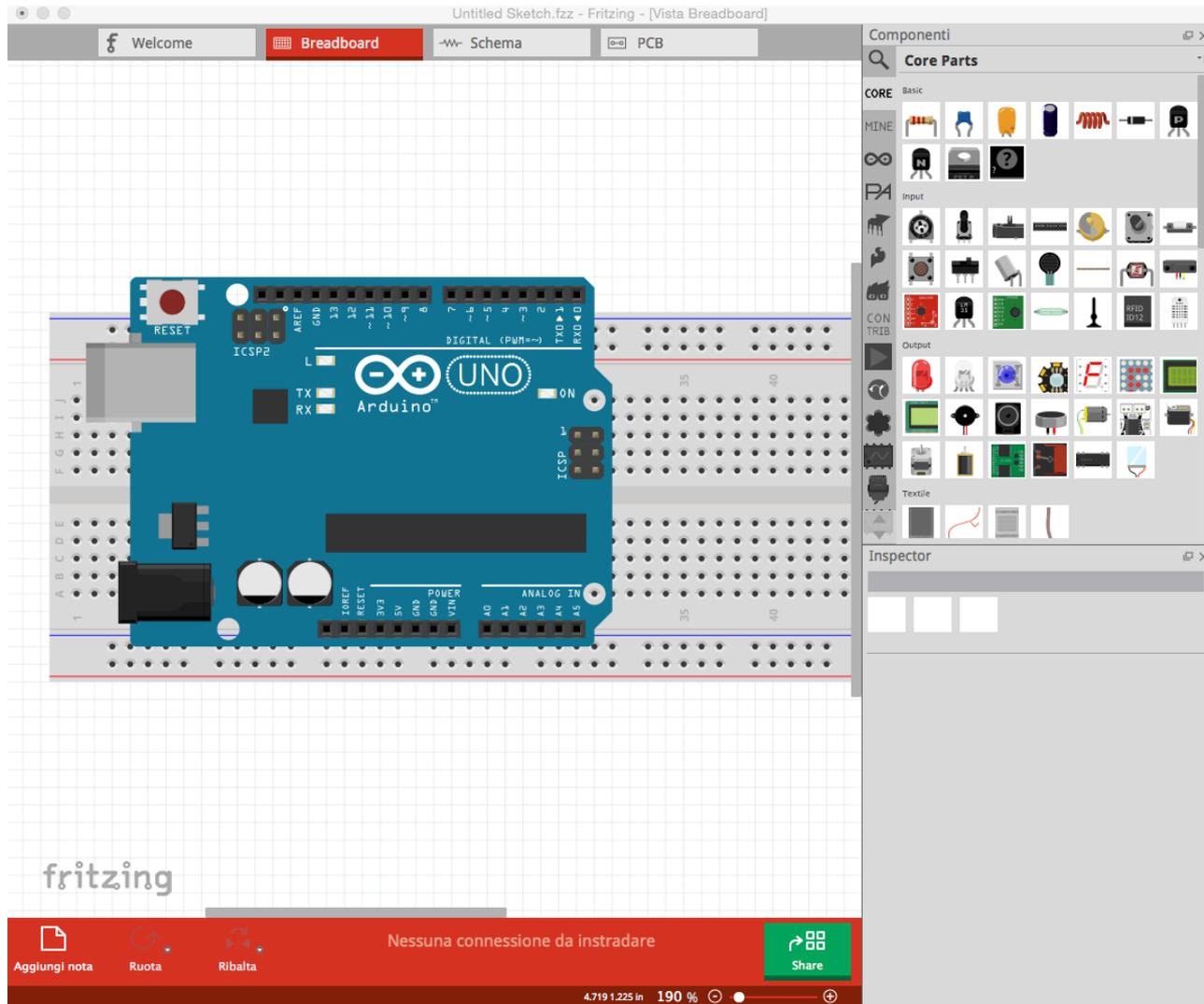
Fritzing è un'iniziativa open-source nata per supportare i progettisti, artisti, ricercatori e appassionati che desiderano realizzare dispositivi elettronici interattivi.

Il software che è in continuo sviluppo ed ampliamento permette agli utenti di documentare i loro prototipi, condividerli con altri, insegnare l'elettronica e creare layout pcb. Sul sito di riferimento tantissimi i progetti già pronti resi disponibili dalla comunità.

Un ottimo strumento da utilizzare in laboratorio di elettronica.



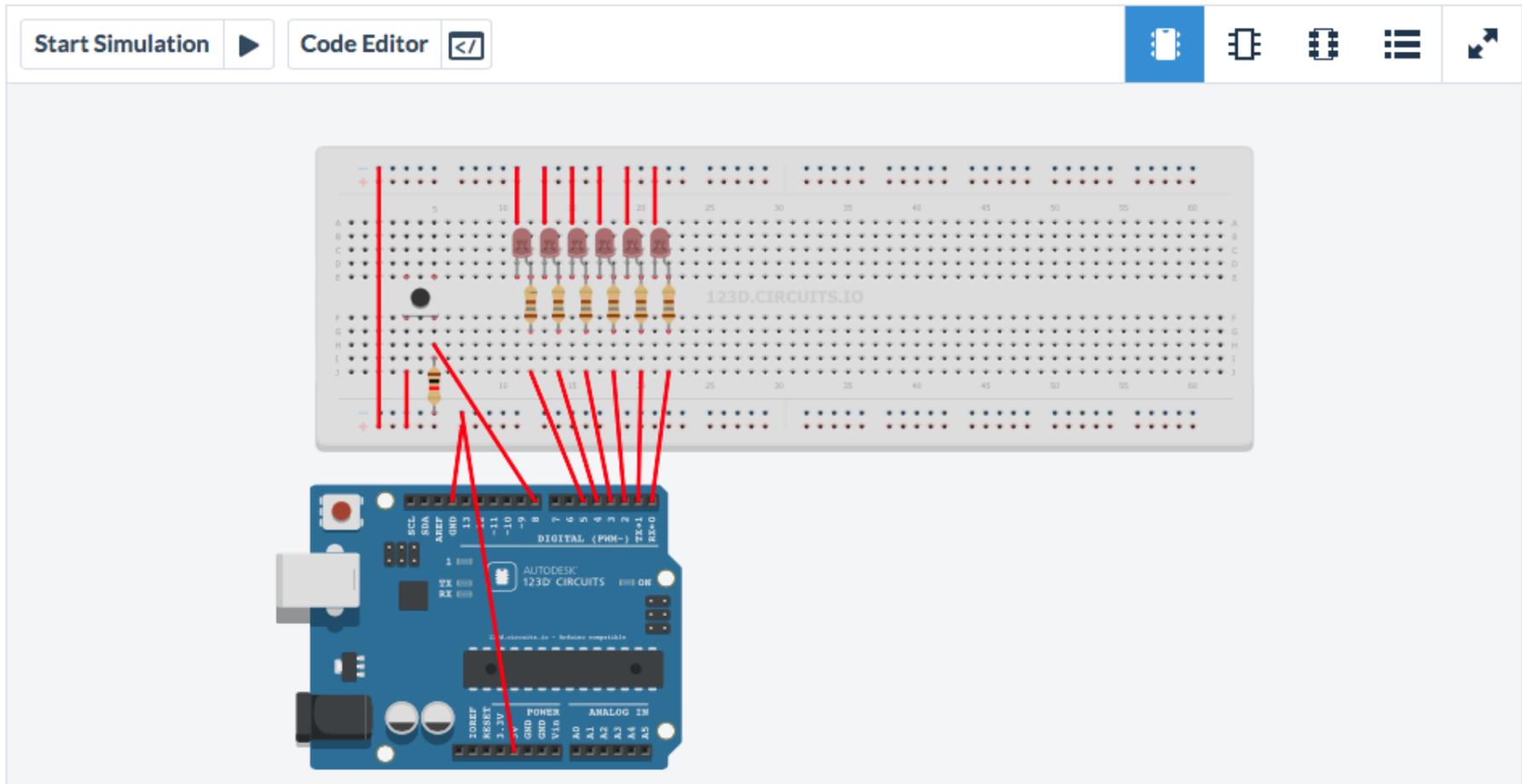
# Fritzing





# 123D Circuits

Design, compile, and simulate  
your electronic project online  
- for free.



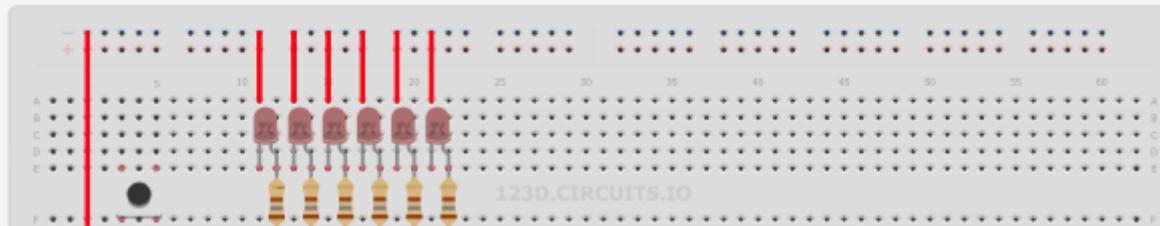
# 123D Circuits

Design, compile, and simulate  
your electronic project online  
- for free.

Start Simulation



Code Editor



1 (Arduino UNO) ▾

Upload &amp; Run



Download Code



Serial Monitor



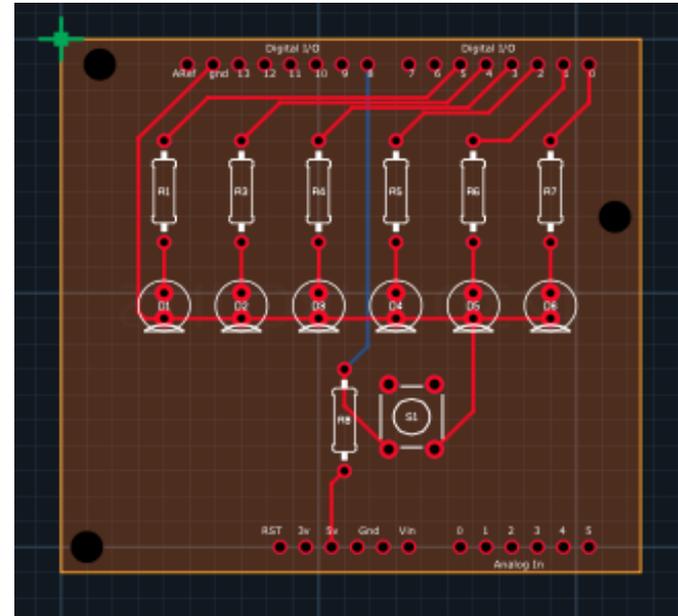
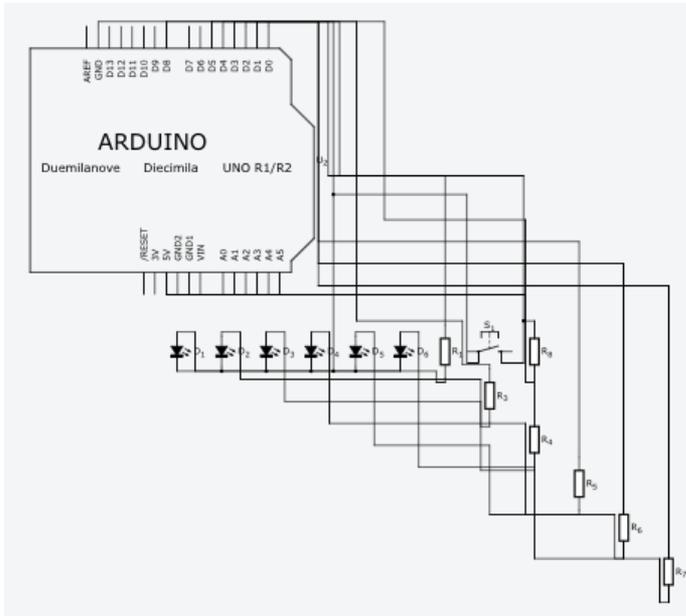
```

1 // the setup routine runs once when you press reset:
2 void setup() {
3   // initialize the digital pin as an output.
4   pinMode(0, OUTPUT);
5   pinMode(1, OUTPUT);
6   pinMode(2, OUTPUT);
7   pinMode(3, OUTPUT);
8   pinMode(4, OUTPUT);
9   pinMode(5, OUTPUT);
10
11   pinMode(8, INPUT);
12 }
13
14 int led =0;
15 // the loop routine runs over and over again forever:
16 void loop() {
17   if (digitalRead(8)) {
18     for(int i=0; i<6;i++) {
19       if(i==led) {
20         digitalWrite(i,HIGH);
21       }

```

# 123D Circuits

Design, compile, and simulate your electronic project online for free.



Download CSV



IDs	#	Component	Footprint	Properties
U2	1	Arduino Uno Rev3		
S1	1	Pushbutton Momentary switches that close a circuit when pressed.	TACTILE-PTH	
D5, D4, D3, D2, D1, D6	6	LED	LED5MM	650 nm
R6, R7, R5, R4, R3, R1	6	resistor	0207/10	180 ohm
R8	1	resistor	0207/10	1 kohm

# Codebender

online development & collaboration platform for Arduino users, makers and engineers

Codebender è un IDE che vi permette di programmare in cloud la vostra scheda Arduino all'interno di una finestra browser.

Utilizzando Codebender potrete scrivere, fare l'upload o il download degli sketch prodotti, compilare ed eseguire il codice di tutte le schede Arduino, compreso l'uso della serial monitor.

Utile la possibilità di condividere il codice con altri utenti o capire l'avanzamento di un progetto, quindi particolarmente adatto in ambito didattico.

All'interno del vostro account potrete visualizzare tutte le schede supportate (tantissime) o aggiungerne di vostre o addirittura fare il burn del bootloader.

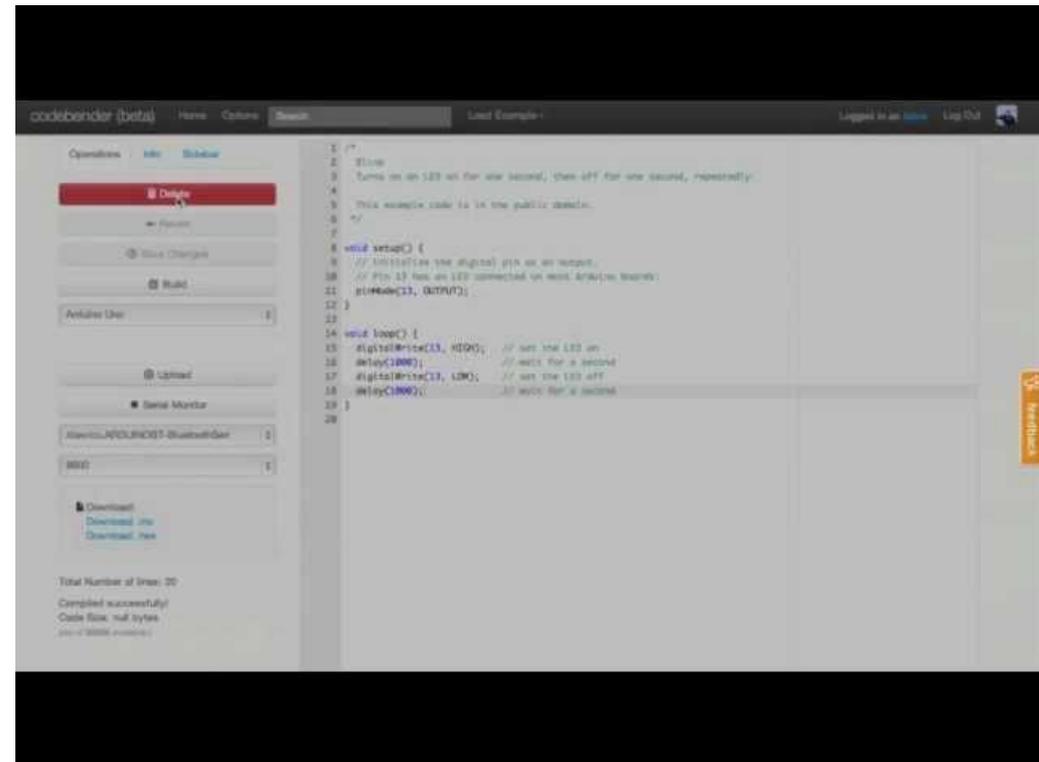
Nella sezione "Examples & Libraries" potete disporre di una quantità incredibilmente vasta di esempi e librerie.

# Codebender

online development & collaboration platform for Arduino users, makers and engineers

Tutorial in italiano su maffucci.it

[Sempre più facile programmare Arduino con CodeBender](#)



**CircuitLab**Effortless schematics.  
Powerful simulation.

Simulatore di circuiti elettronici (analogici e digitali) on-line.

Potrete verificare rapidamente i vostri progetti prima di effettuare la costruzione su breadboard o su millefori.

Caratteristiche principali:

- interfaccia assolutamente user friendly,
- facilità di connessione tra i vari componenti,
- possibilità di rielaborare circuiti pubblicati da altri utenti,
- possibilità di simulare nello stesso progetto circuiti costituiti da componenti digitali ed analogici,
- disegno di grafici
- i circuiti possono essere condivisi o esportati in diversi formati.

**New Public Circuits**

2 minutes ago

- Linear DC Power Supply (Zener Reg)
- Design project 1
- FloydKingDesign1

**Active Forum Discussions**

- Old MOSFET symbol
- Include UJT- PUT in components
- Are There Any Jobs For Electronics...

**Latest Blog Posts**

- Server Availability Improvements
- Expanded Device Library
- Custom Device Models and SPICE...

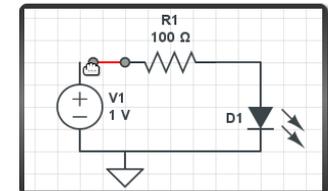
**Quick-Start Circuits**

New to CircuitLab? Jump right in to one of our sample circuits:

- 555 Timer as Oscillator / PWM Generator
- 7805 & Wall-wart Experiencing Voltage Drop-out
- BJT audio amplifier
- BJT Cascoded Active-load Diff. Amp. with CMFB
- BJT current mirror
- Digital 4-bit counter and DAC
- Diode half-wave rectifier
- Diode full-wave (bridge) rectifier
- Diode turn-off time
- Laplace transform step response and Bode plot
- LED with resistor biasing
- JFET-based electret microphone amplifier
- Mechanical spring dashpot Laplace model

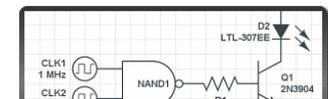
**Easy-to-use Power Tools**

**Easy-wire mode** lets you connect elements with fewer clicks and less frustration.



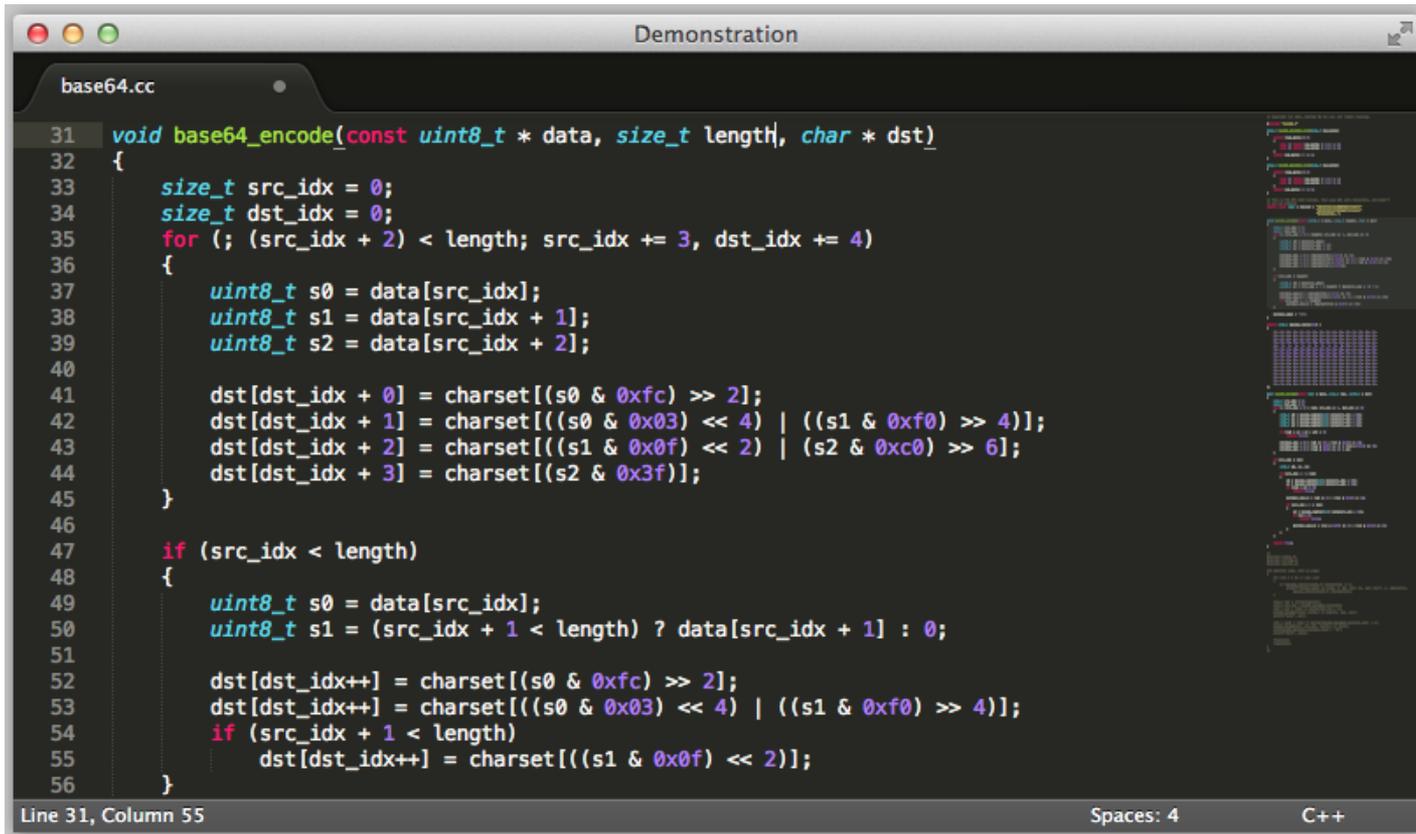
**Cross-window copy/paste** lets you easily explore and re-mix parts of public circuits from the CircuitLab community.

**Mixed-mode circuit simulation** lets you simulate analog and digital components side-by-side.



# Sublime Text

Sublime Text is a sophisticated text editor for code, markup and prose. You'll love the slick user interface, extraordinary features and amazing performance.



```
31 void base64_encode(const uint8_t * data, size_t length, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < length; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < length)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < length) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < length)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
```

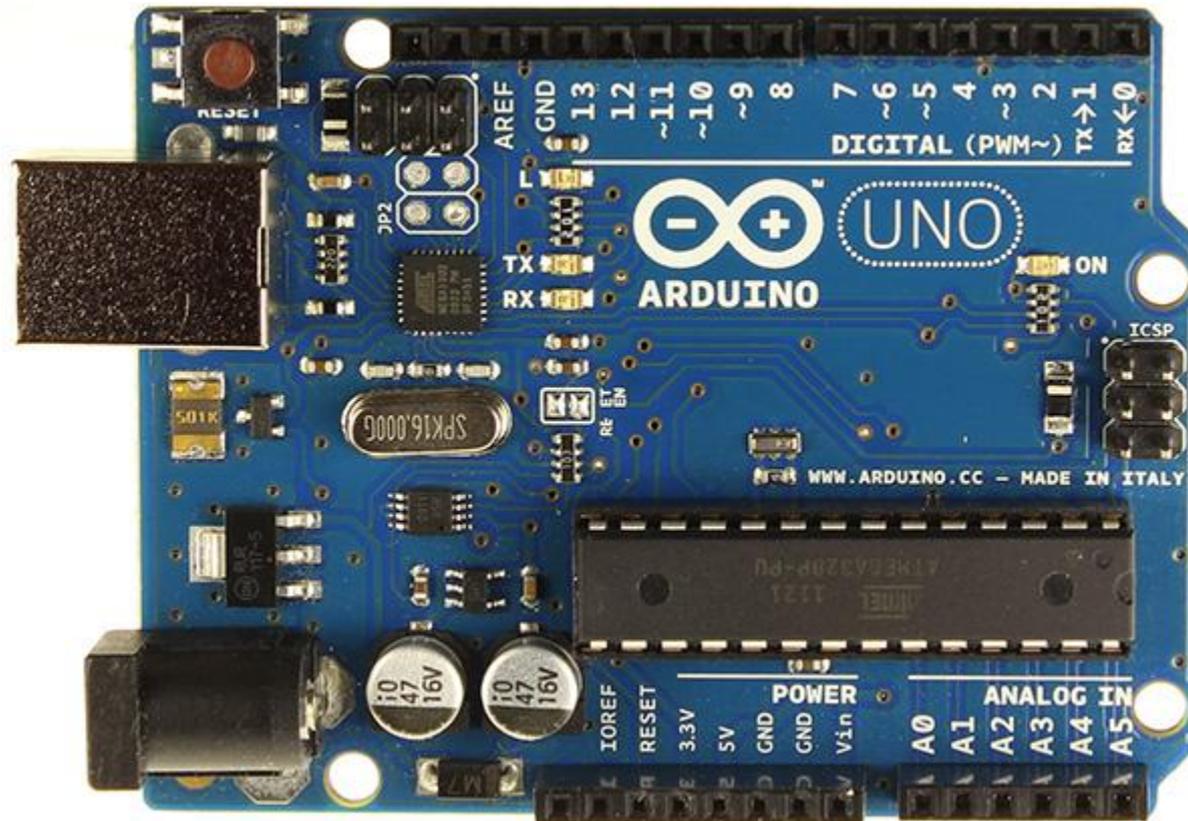
Tutorial in italiano su [maffucci.it](http://maffucci.it)

[Utilizzare Sublime Text 3 Editor come IDE Arduino](#)

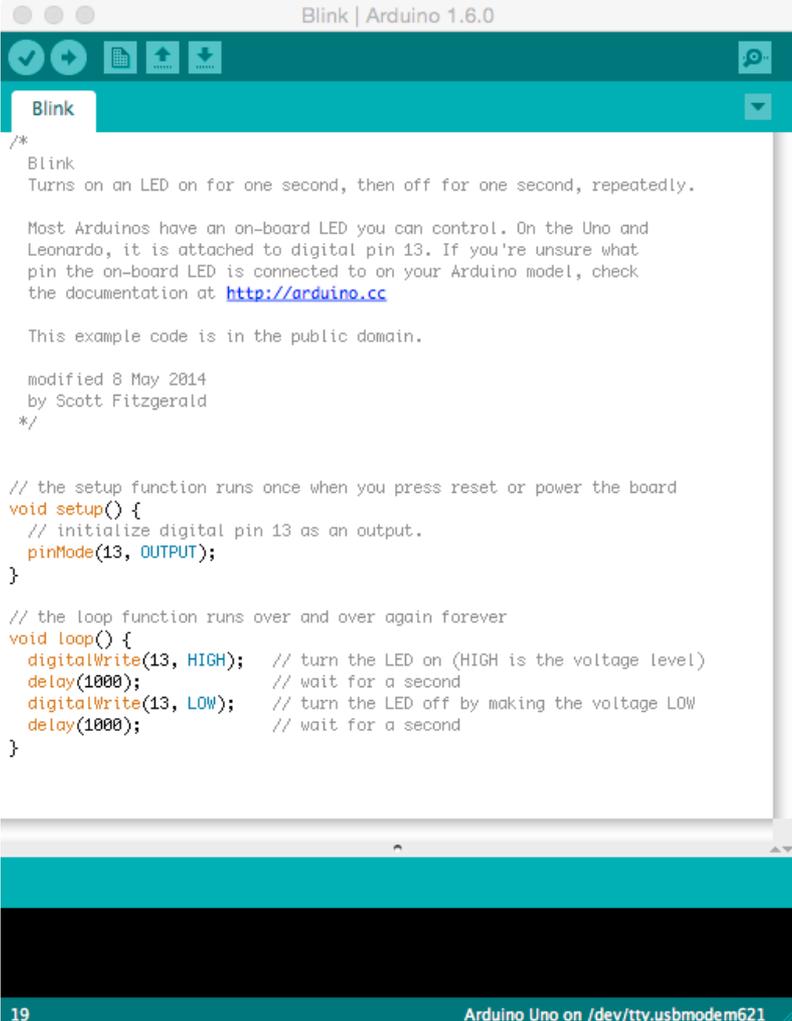
# **Universo Arduino**

# Arduino vuol dire 3 cose

# Un oggetto fisico



# un'ambiente di sviluppo (di programmazione)



The screenshot shows the Arduino IDE interface. The title bar reads "Blink | Arduino 1.6.0". The menu bar includes "Blink" and a dropdown arrow. The main text area contains the following code:

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the Uno and
Leonardo, it is attached to digital pin 13. If you're unsure what
pin the on-board LED is connected to on your Arduino model, check
the documentation at http://arduino.cc

This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}
```

At the bottom of the IDE, a status bar shows "19" on the left and "Arduino Uno on /dev/tty.usbmodem621" on the right.

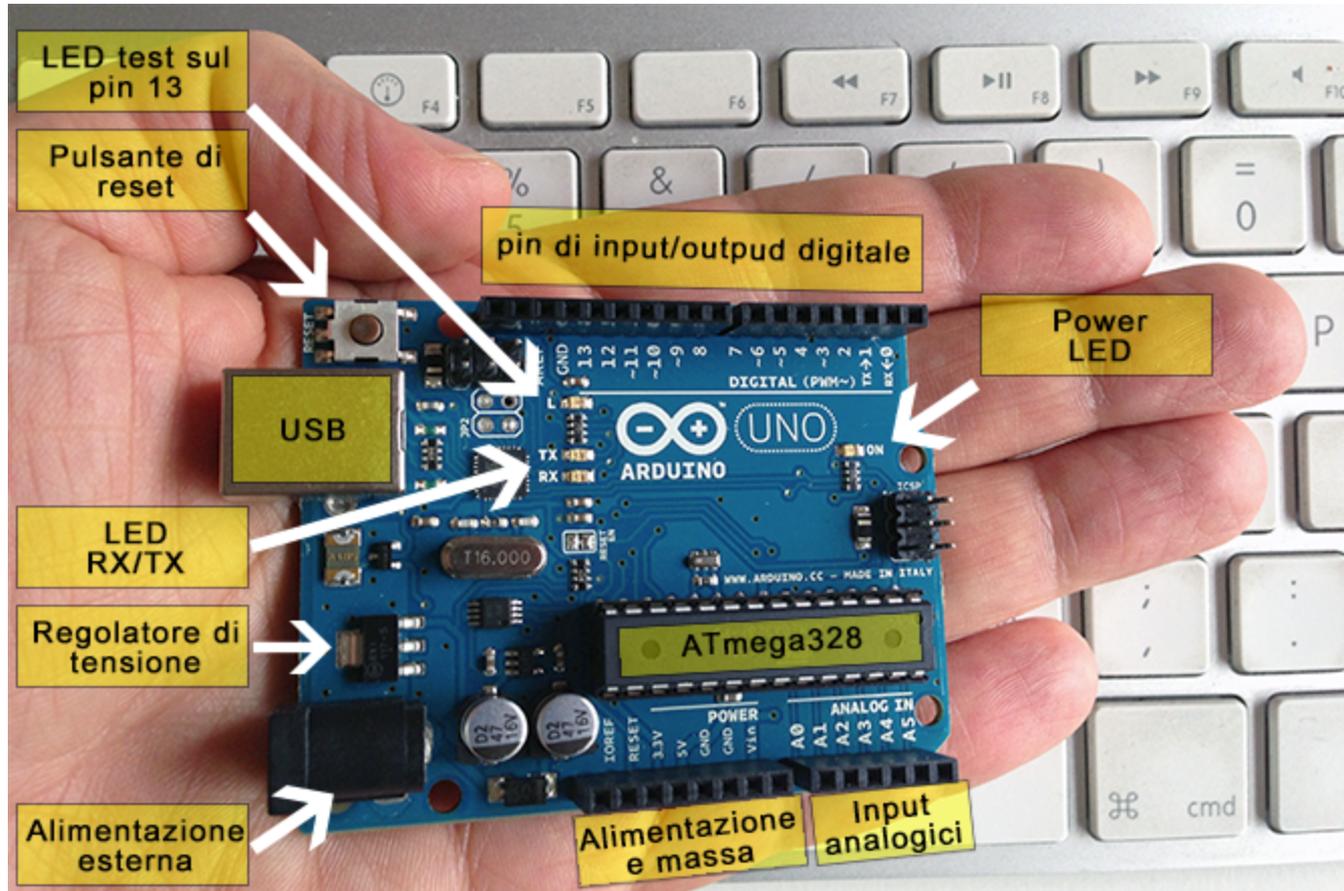
# una comunità ed una filosofia di sviluppo

The collage features several screenshots of the Arduino website:

- Getting Started with Arduino:** Shows the main navigation menu (Home, Buy, Download, Products, Learning, Forum, Support, Blog) and the 'Getting Started with Arduino' page. The page includes sections for 'Introduction: What Arduino is and why you'd want to use it.' and 'Installation: Step-by-step instructions for setting up the Arduino software... connecting it to an Arduino Uno, Mega2560, Duemilanove, Mega, or Diecimila'.
- Examples:** Shows a sidebar with categories like 'Using Arduino', 'Installation & Troubleshooting', 'Project Guidance', 'Programming', and 'General Electronics'. The main content area lists 'Examples | Foundations | Hardware | Reference | Language | Libraries | Comparison | Changes'.
- Language Reference:** Shows the 'Language Reference' section with a note: 'NOTE: THESE EXAMPLES ARE WRITTEN FOR Arduino programs can be divided in three main parts: *structure*, *values* (variables) and *functions*.'
- The Arduino Playground:** Shows the 'The Arduino Playground' page with a welcome message: 'Welcome to the Arduino Playground, a wiki where a community of people can contribute and benefit from their collective research and knowledge. This is the place to post and share your own code, circuit diagrams, DIY instructions, tips and tricks, and after all the hard work, help other people's projects! Anyone can edit and add to the pages here.' It also includes a 'Participate' section and a 'Playground Content Tree'.

# **La scheda Arduino**

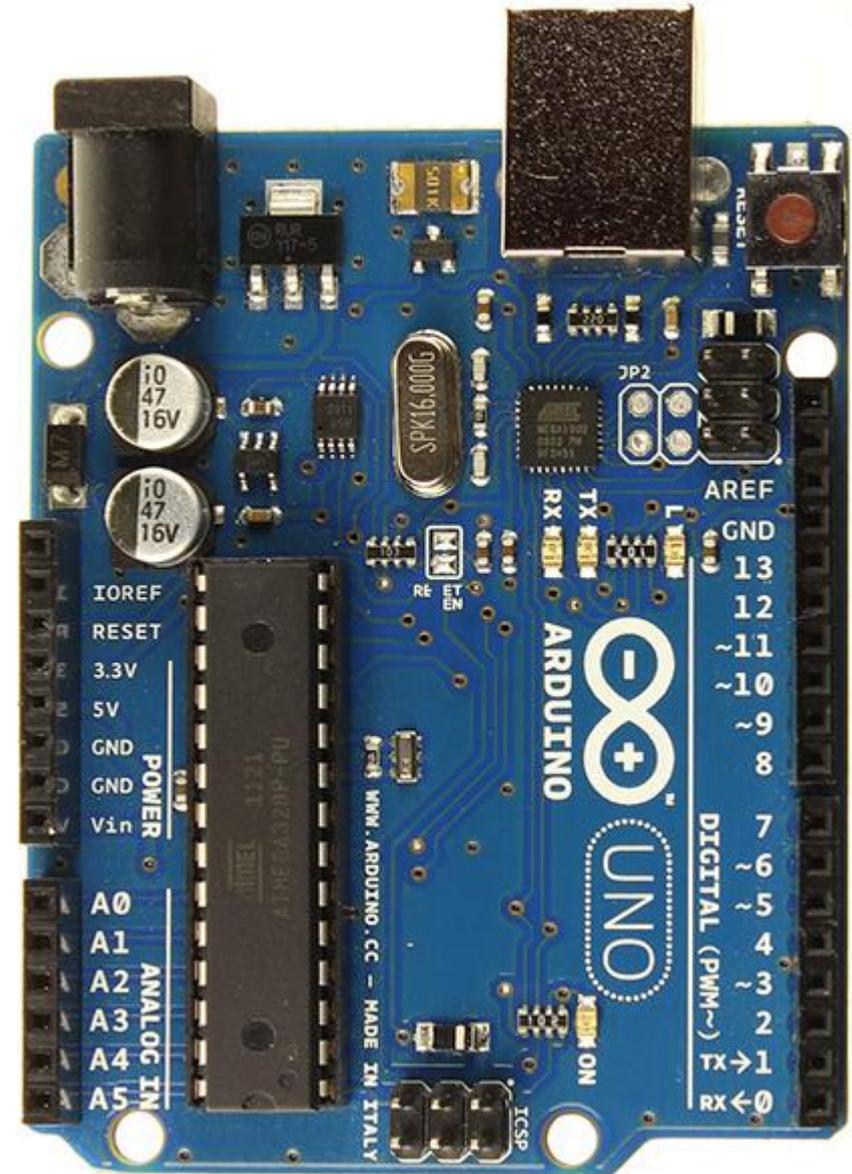
Elementi di base



Arduino UNO R3

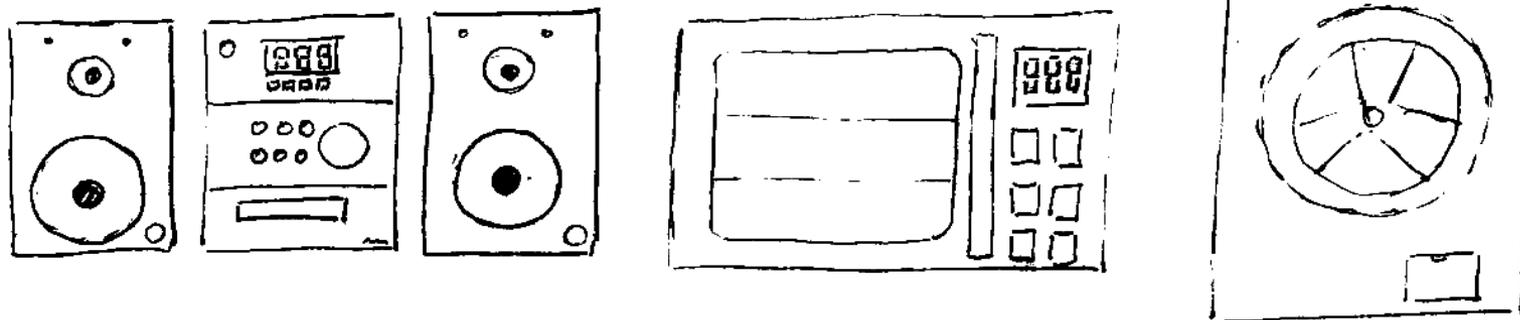
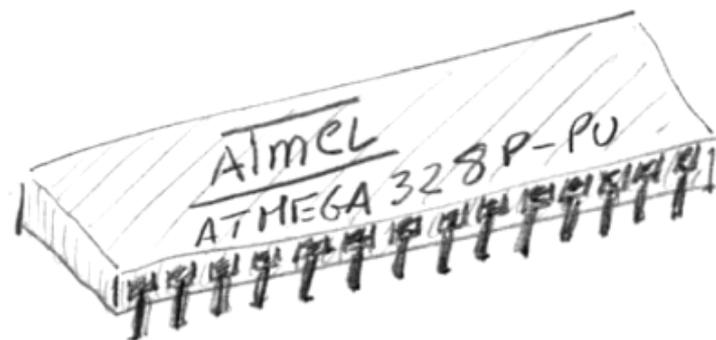
## Caratteristiche tecniche

- Microcontroller: ATmega328
- Tensione di lavoro: 5V
- Tensione di ingresso (raccomandata): 7-12V
- Tensione di ingresso (limiti): 6-20V
- Pin digitali I/O: 14 (di cui 6 forniscono un'uscita PWM)
- Pin analogici: 6
- Corrente Continua per i pin I/O: 40 mA
- Corrente continua per l'uscita a 3.3V: 50 mA
- Flash Memory: 32 KB (ATmega328) di cui 0.5 KB usata per bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Velocità del clock: 16 MHz



## Il microcontrollore

Il cuore della scheda Arduino è il **microcontrollore**, un dispositivo elettronico molto simile ad un computer in miniatura che potete trovare in molti degli elettrodomestici che usate ogni giorno: lavatrice, cellulare, forno a microonde, impianto HiFi, ecc...



E' molto probabile che se l'elettrodomestico possiede pulsanti e display e rileva grandezze fisiche (temperatura, pressione, ecc...) abbia al suo interno un microcontrollore.

# **Terminologia essenziale**

## sketch

il programma che scrivete e fate girare sulla scheda Arduino

## pin

i connettori di input o output

## digital

vuol dire che può assumere solo due valori: ALTO o BASSO, in altro modo: ON/OFF oppure 0 o 1. Sequenza di numeri presi da un insieme discreto di valori (nel nostro caso 0 o 1)

## analog

quando i valori utili che rappresentano un segnale sono continui (infiniti)

# **Il software Arduino**

## Il software

L'ambiente di sviluppo viene comunemente chiamato Arduino e ciò può trarre in confusione, perché si identifica con Arduino anche la scheda hardware.

In queste lezioni per indicare l'ambiente di sviluppo software useremo le parole:

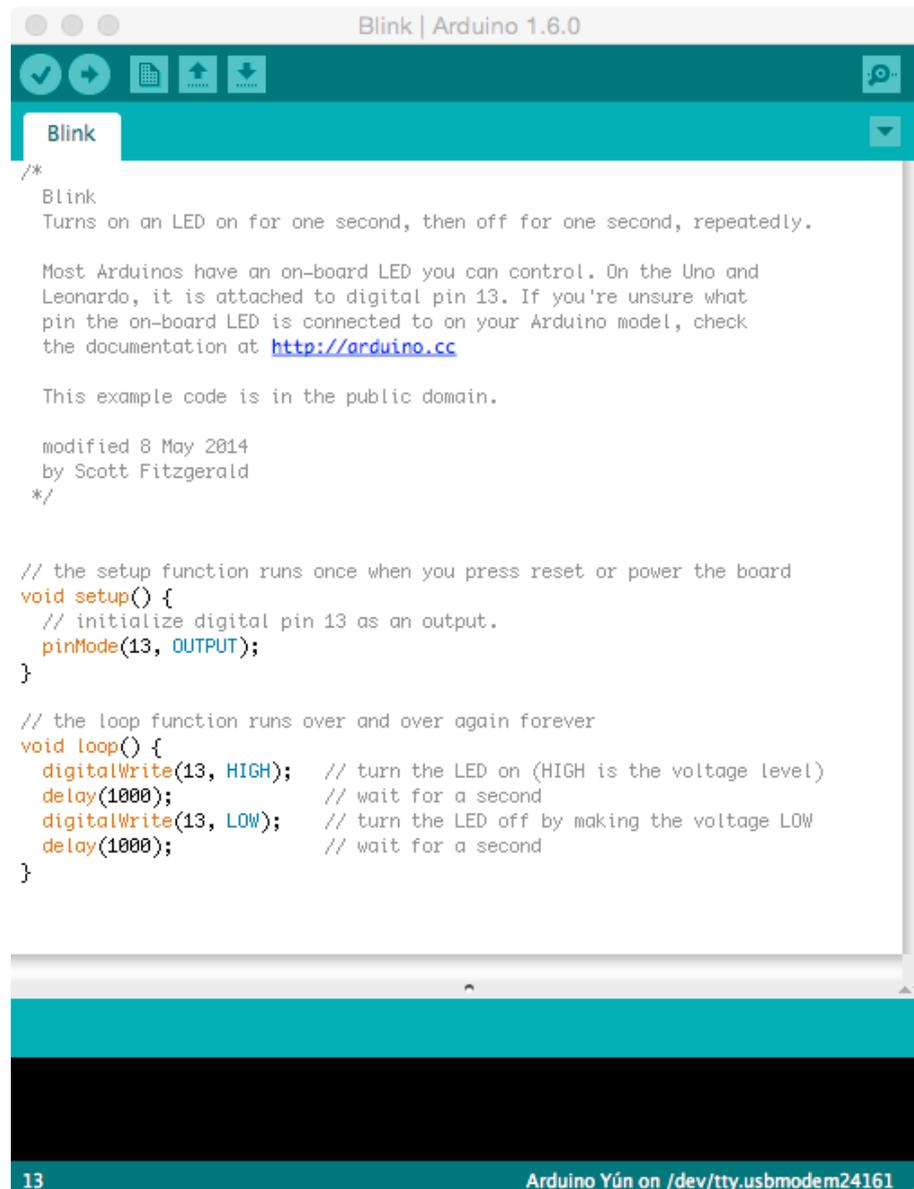
## software Arduino

o con stesso significato

## IDE

dove l'acronimo **IDE** indica: *Integrated Development Enviroment*,  
in italiano: *ambiente di sviluppo integrato per la realizzazione di programmi*.

## Il software



The screenshot shows the Arduino IDE interface with the 'Blink' sketch loaded. The code is as follows:

```
Blink | Arduino 1.6.0
Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

13 Arduino Yún on /dev/tty.usbmodem24161
```

- Simile ad un editor di testo;
- potete scrivere, visualizzare, verificare la sintassi;
- potete trasferire il vostro sketch sulla scheda.

1. prelevare il software Arduino dal sito [arduino.cc](http://arduino.cc)
2. collegare la scheda Arduino al computer
3. installare i driver
4. riavviare il computer
5. avviare il software Arduino
6. scrivere uno sketch
7. eseguire lo sketch facendo l'upload sulla scheda Arduino

Il software

collegarsi al sito arduino.cc



Search the Arduino Website

Home Buy Download Products Learning Forum Support Blog



### WHAT IS ARDUINO?



BUY AN ARDUINO



LEARN ARDUINO



### ARDUINO ATHEART



Designed for makers and companies wanting to make their products easily recognizable as based on the Arduino technology.

SHOW OFF YOUR



BLOG

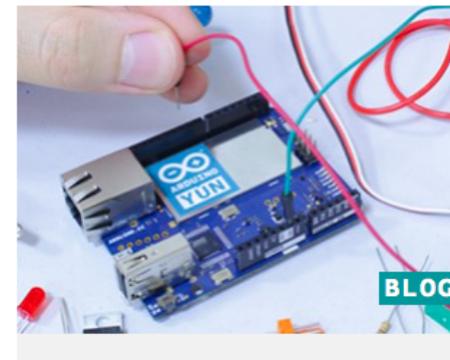
DECONSTRUCTING IOT:  
TEMBOO VIDEO-INTERVIEWS  
TOM IGOE



ARDUINO  
DAY 2015  
MARCH 28<sup>TH</sup>



IoT / DESIGN WORKSHOP  
FEBRUARY 21-22  
TORINO, ITALY  
SIGN UP!



BLOG

Il software

Download



Search the Arduino Website



Home

Buy

Download

Products ▾

Learning ▾

Forum

Support ▾

Blog



## Download the Arduino Software



### ARDUINO 1.6.0

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows** Installer**Windows** ZIP file for non admin install**Mac OS X** for Java 6 (recommended)**Mac OS X** for Java 7 (experimental)**Linux** 32 bits**Linux** 64 bits[Release Notes](#) [Source Code](#)

ARDUINO 1.0.x / 1.5.x

### PREVIOUS RELEASES

Download the Arduino 1.0.6 and all the previous versions of the Arduino Software. Available for Windows, Linux, and Mac OS X.

ARDUINO IDE

### INTEL GALILEO AND EDISON

Download the Arduino IDE that supports the Intel Galileo and the Intel Edison boards. Available for Windows, Linux, and Mac OS X.

Il software

Download



Buy

Download

Products ▾

Learning ▾

Forum

Support ▾

Blog



## Nightly Builds

Download a preview of the incoming release with the most updated features and bugfixes.

Windows

Mac OS X

Linux 32 bit, Linux 64 bit

## Source Code

Active development of the Arduino software is hosted by GitHub. See the instructions for [building the code](#). Source code of Arduino is available [here](#).

## Other Software

### ARDUINO YÚN LINUX OS OPENWRT-YÚN 1.5.3 UPGRADE IMAGE

Download the latest stable GNU/Linux OS for your Yún. It makes your Yún more stable and feature rich.

**Easy Installation Procedure (recommended):** Download the Upgrade Image then please follow the steps in the [Yún sysupgrade tutorial](#).

**Advanced Installation Procedure:** This procedure is only recommended to advanced users who wish to completely re-flash the Yún including its U-Boot bootloader. [These instructions](#) on reflashing the base images are for reference only. Following them will void your Yún's warranty.

### Packages list

The list of available packages for the Yún is available [here](#). See the list of [changes](#).

Il software

installazione

## Windows

[arduino.cc/windows](https://arduino.cc/windows)

installazione per: Windows 7, Vista, e XP

## Mac OS X

[arduino.cc/mac](https://arduino.cc/mac)

installazione per: OS X 10.5 e successive

## Linux

[arduino.cc/linux](https://arduino.cc/linux)

installazione per: disponibile per moltissime distribuzioni Linux

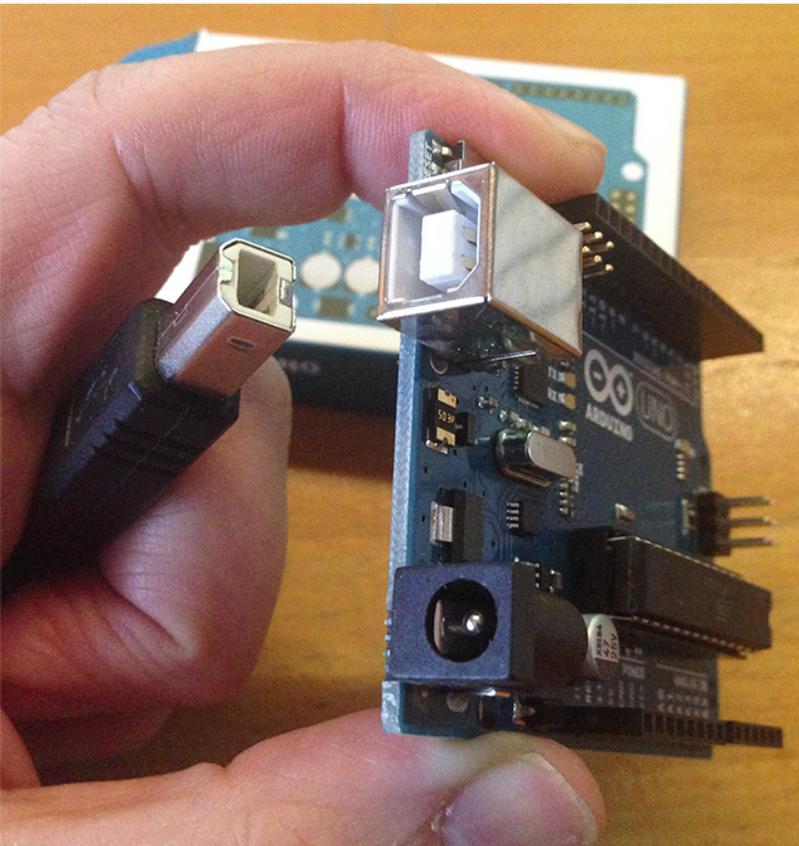
approfondimenti su installazione **Mac** e **Linux** su:[www.maffucci.it/area-studenti/arduino/](http://www.maffucci.it/area-studenti/arduino/)

# **Comunicare con Arduino**

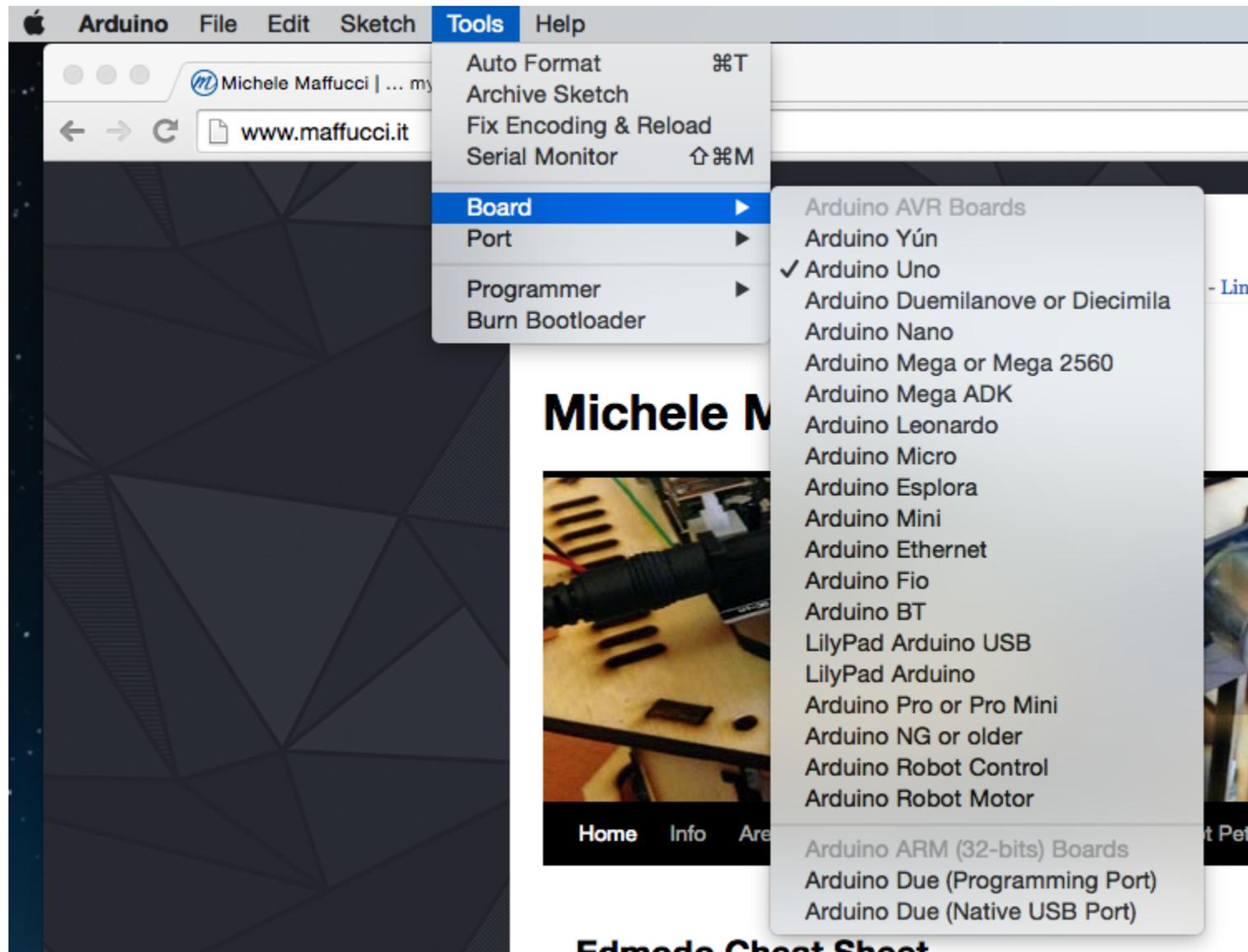
Avviate l'IDE di programmazione facendo doppio click sull'icona di Arduino



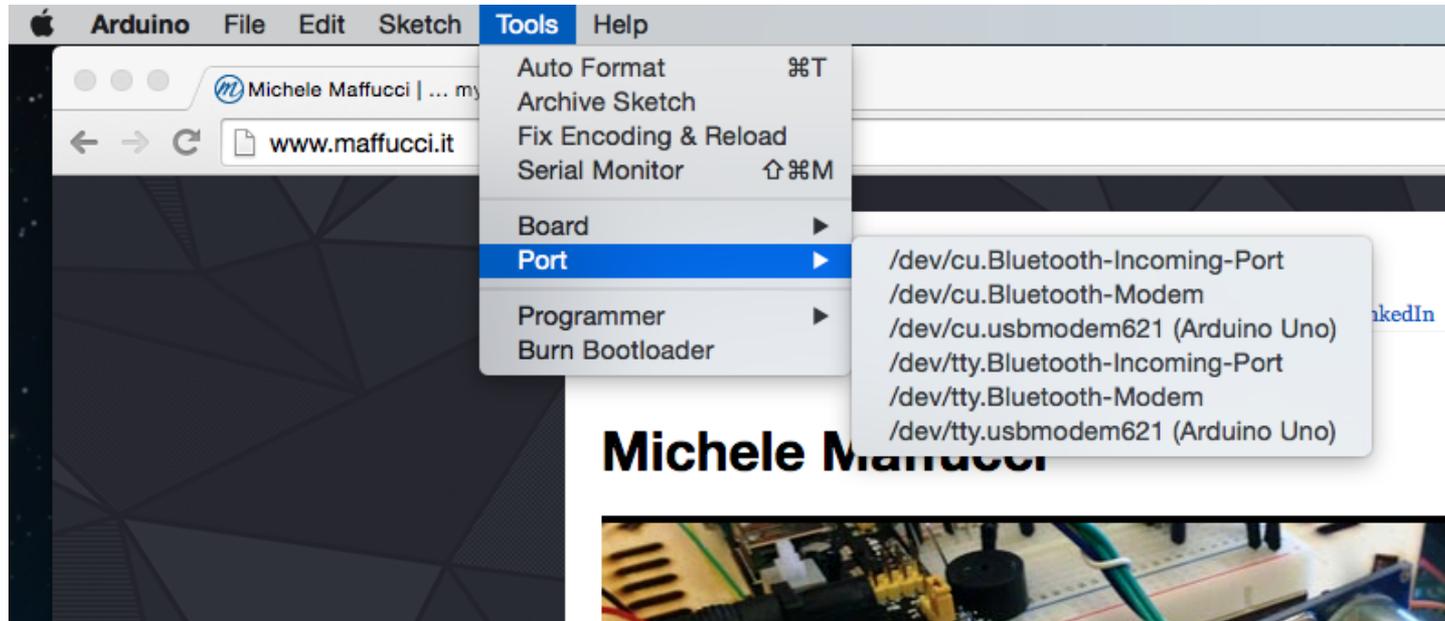
Collegare la scheda Arduino al computer mediante cavo USB (tipo B)



Selezionate la scheda in vostro possesso, nel nostro caso Arduino Uno: **Tools > Board > Arduino Uno**



Selezionate la porta seriale da utilizzare per la comunicazione tra computer ed Arduino: **Tools > Serial port**

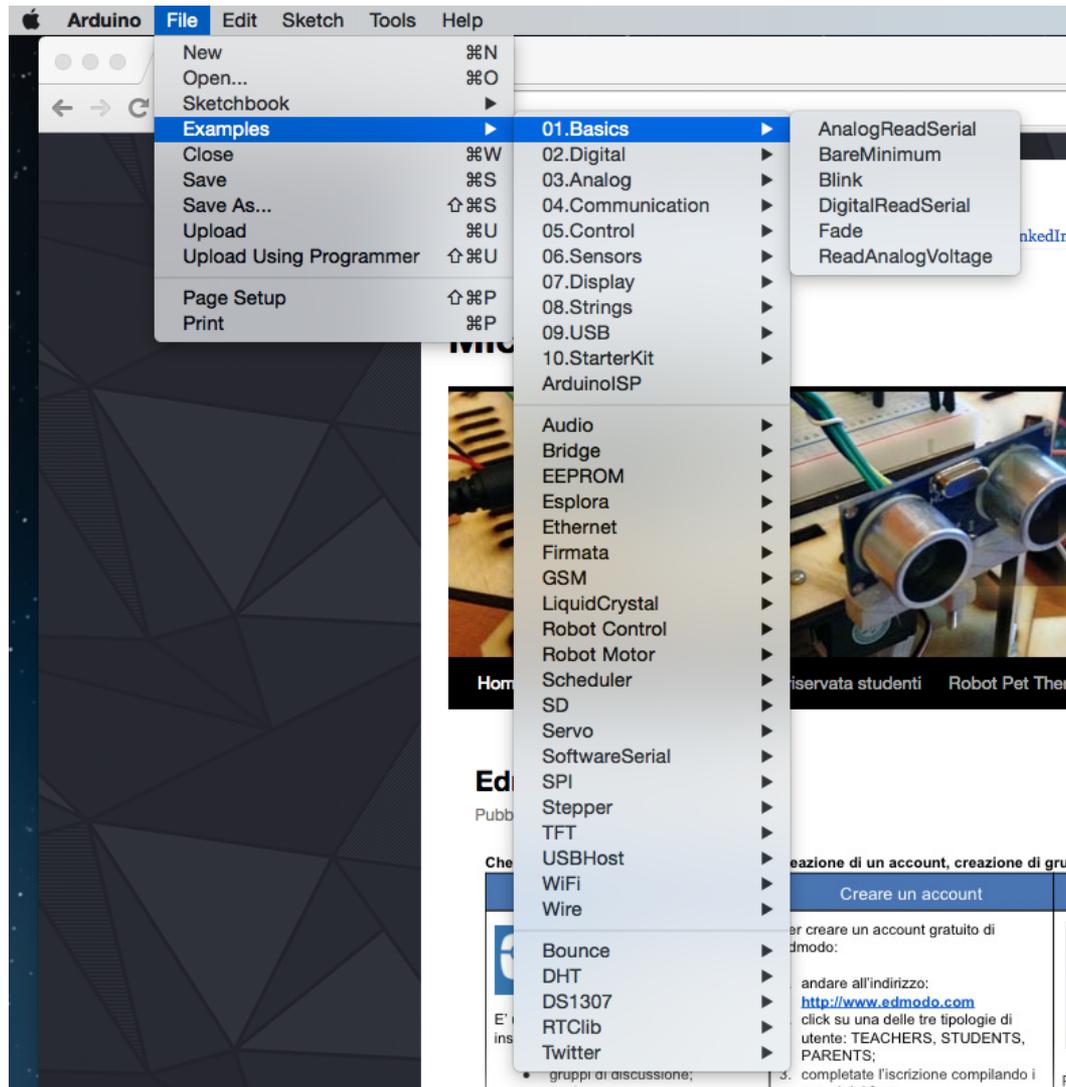


Su **Mac** potete selezionare indifferentemente `/dev/tty.usbmodemXXX` oppure `/dev/cu.usbmodemXXX`

Su **Windows** dovrete notare una o più porte COM, selezionate quella con numero più elevato, se non dovesse funzionare provate con le altre proposte.

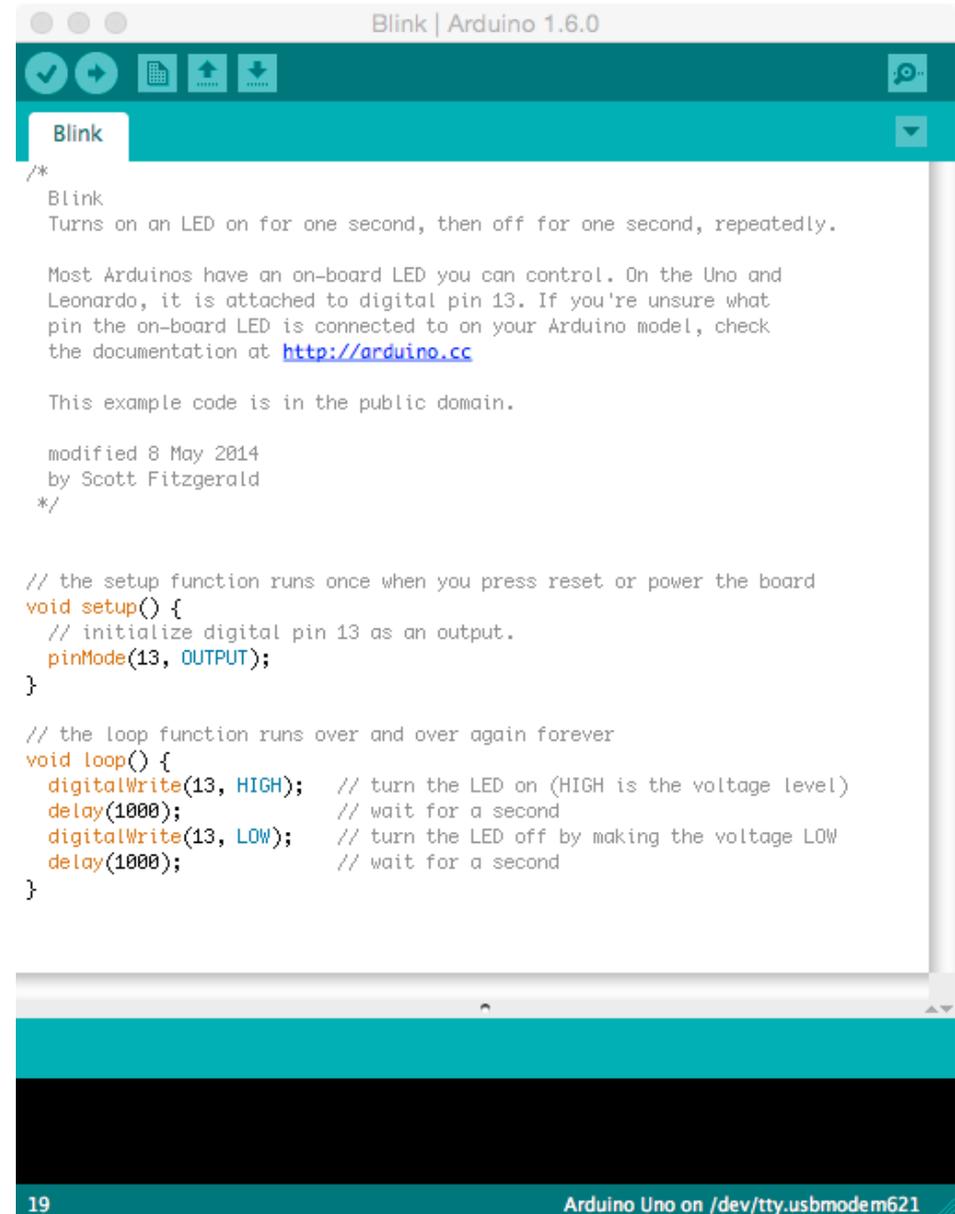
Su **Linux** (Ubuntu) dovrete vedere una `ttyACM0`. Per maggiori informazioni consultare la sezione Arduino su Ubuntu su: [www.maffucci.it/area-studenti/arduino/](http://www.maffucci.it/area-studenti/arduino/)

Aprire lo sketch di esempio **blink** che fa lampeggiare il LED presente sulla scheda. Lo sketch può essere aperto da: **File > Examples > 01. Basics > Blink**



Si aprirà una finestra con il codice del programma blink.

Studieremo più avanti il funzionamento.

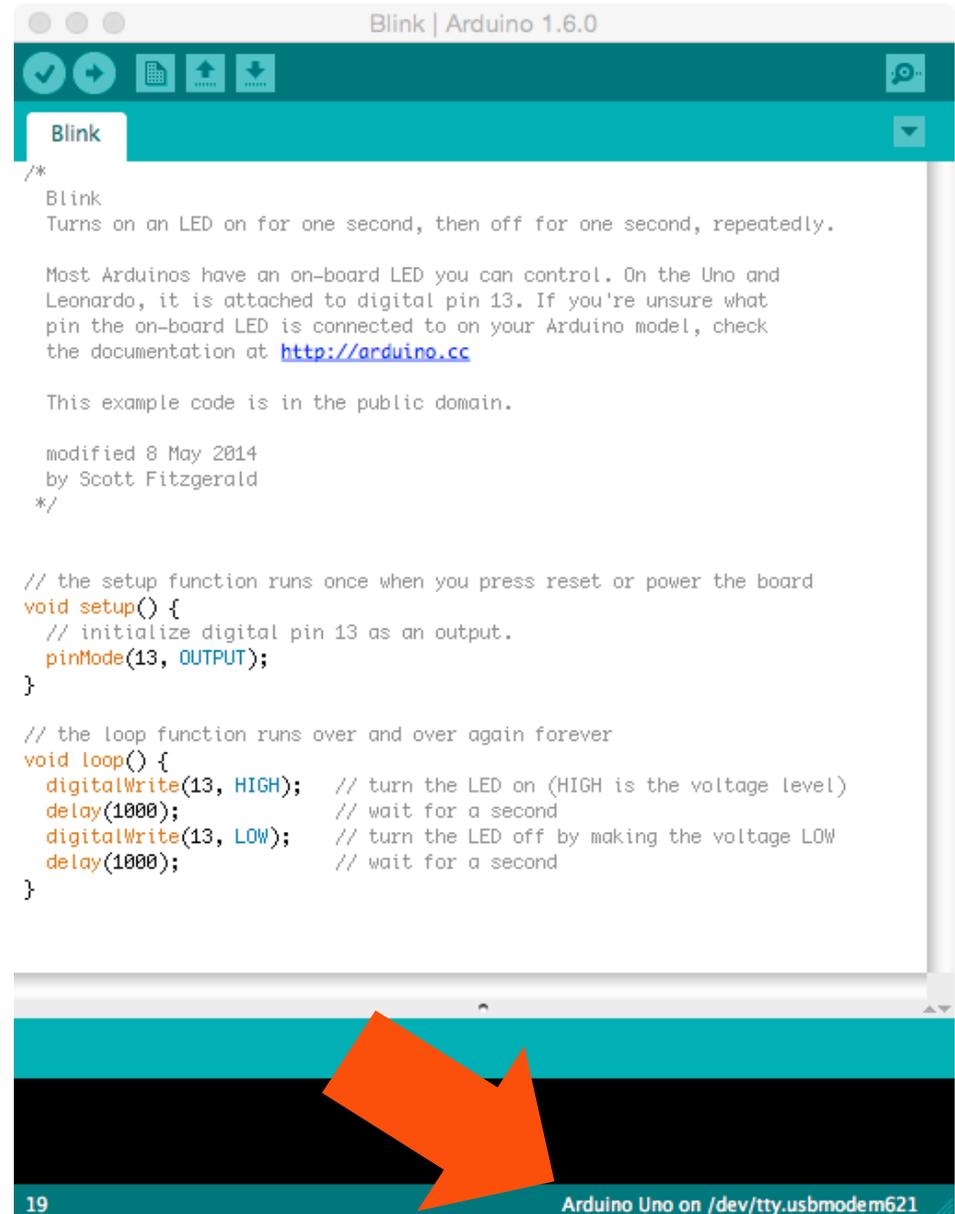
A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.6.0". The interface shows a toolbar with icons for check, back, forward, and refresh. Below the toolbar is a tab labeled "Blink". The main area contains the following code:

```
/*  
Blink  
Turns on an LED on for one second, then off for one second, repeatedly.  
  
Most Arduinos have an on-board LED you can control. On the Uno and  
Leonardo, it is attached to digital pin 13. If you're unsure what  
pin the on-board LED is connected to on your Arduino model, check  
the documentation at http://arduino.cc  
  
This example code is in the public domain.  
  
modified 8 May 2014  
by Scott Fitzgerald  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin 13 as an output.  

```

At the bottom of the window, a status bar shows "19" on the left and "Arduino Uno on /dev/tty.usbmodem621" on the right.

Il collegamento alla porta seriale viene segnalato nella finestra del codice in basso a destra



```
Blink | Arduino 1.6.0
Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://arduino.cc

  This example code is in the public domain.

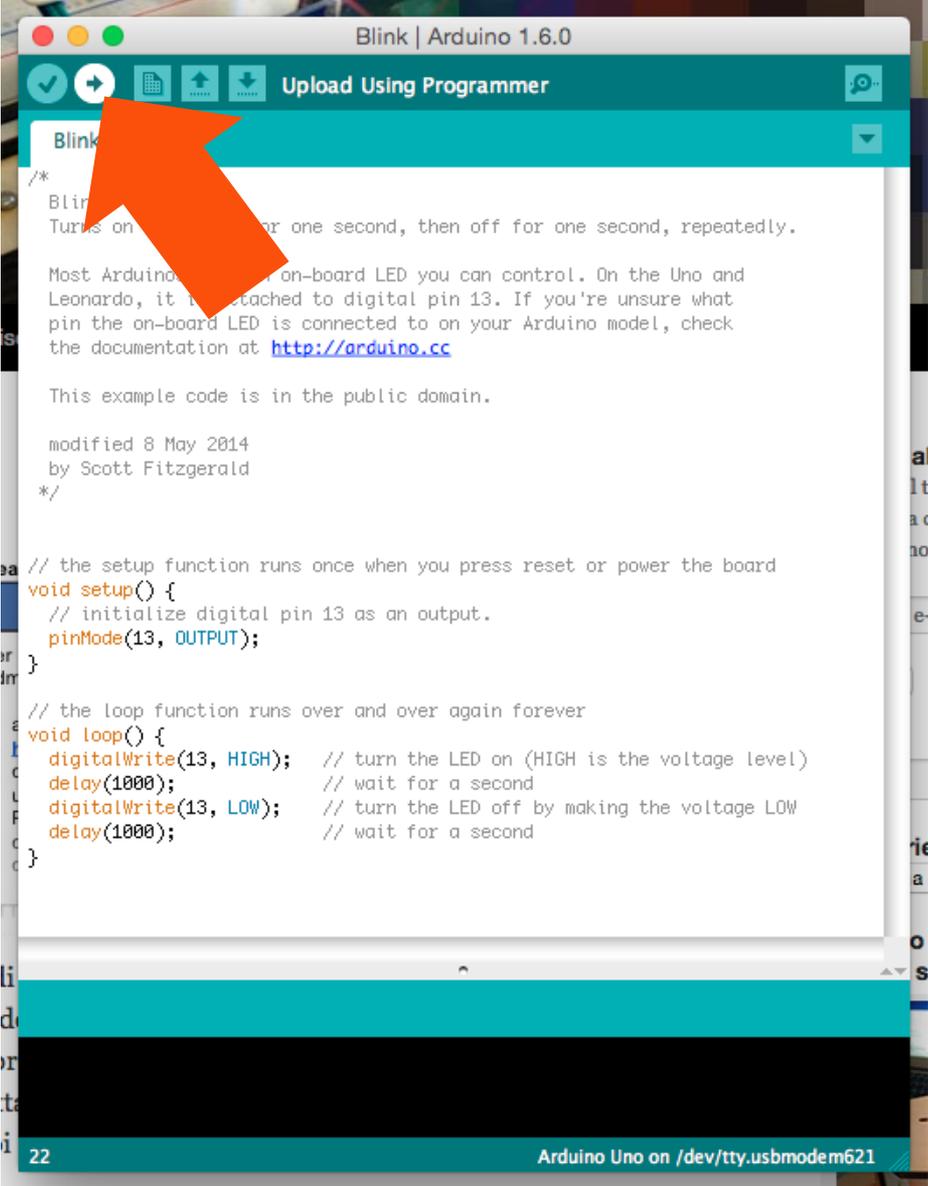
  modified 8 May 2014
  by Scott Fitzgerald
  */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}

19 Arduino Uno on /dev/tty.usbmodem621
```

Procedere con il caricamento dello sketch Blink sulla scheda mediante il pulsante Upload nella finestra in cui compare il codice:



```
Blink | Arduino 1.6.0
Upload Using Programmer
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. On the Uno and
 * Leonardo, it is attached to digital pin 13. If you're unsure what
 * pin the on-board LED is connected to on your Arduino model, check
 * the documentation at http://arduino.cc
 *
 * This example code is in the public domain.
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

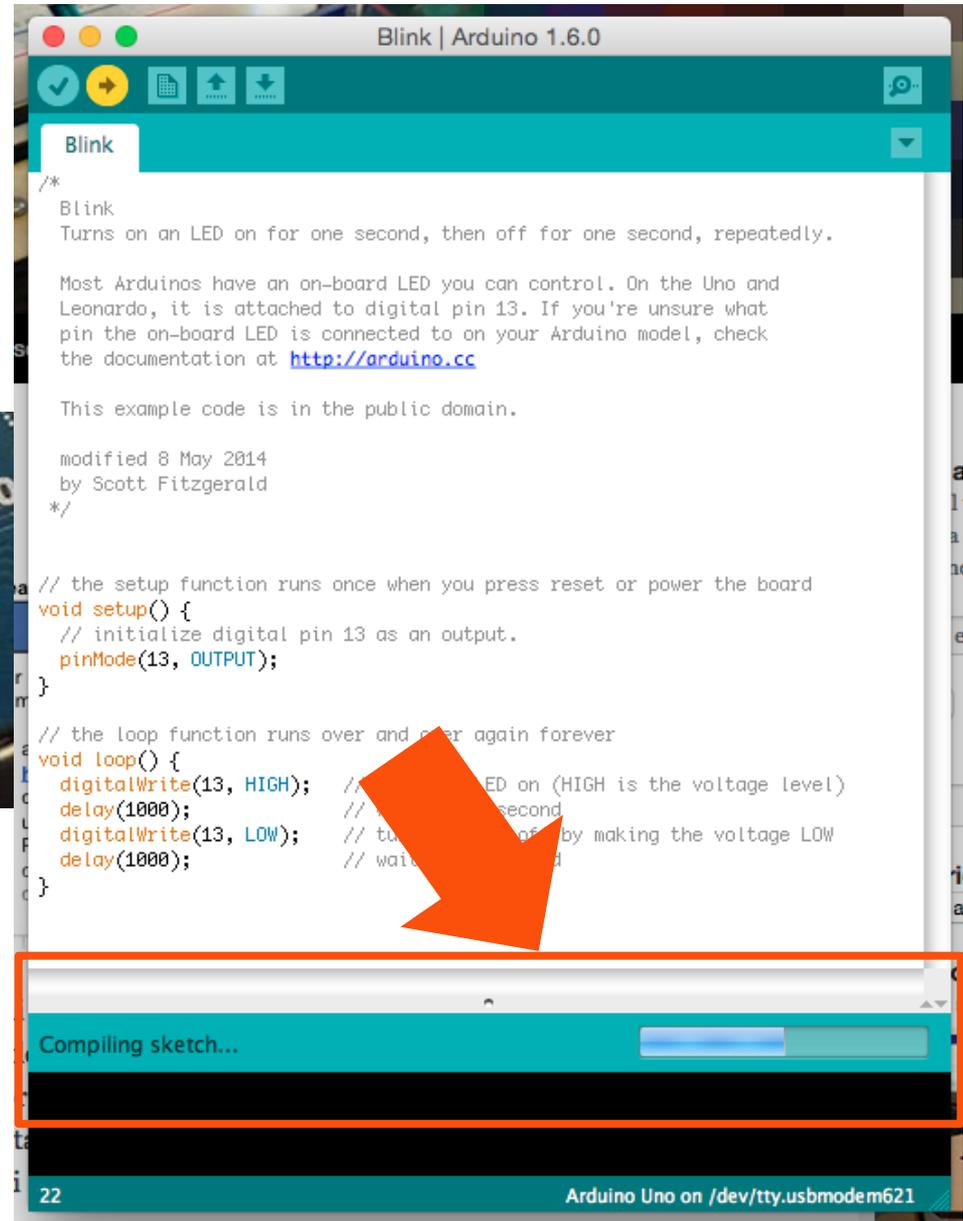
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

22 Arduino Uno on /dev/tty.usbmodem621
```

## aprire sketch blink

9/11

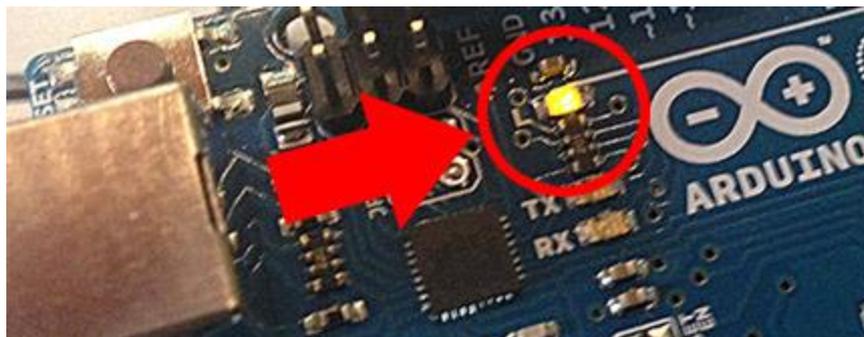
Ci vorrà qualche secondo, durante questa operazione vedrete che i led RX e TX (ricezione e trasmissione) lampeggiano.



aprire sketch blink

10/11

Se tutto andrà a buon fine vi verrà restituito il messaggio "Done uploading." nella staus bar ed il LED L incomincia a lampeggiare



```

Blink | Arduino 1.6.0
Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
  */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}

```

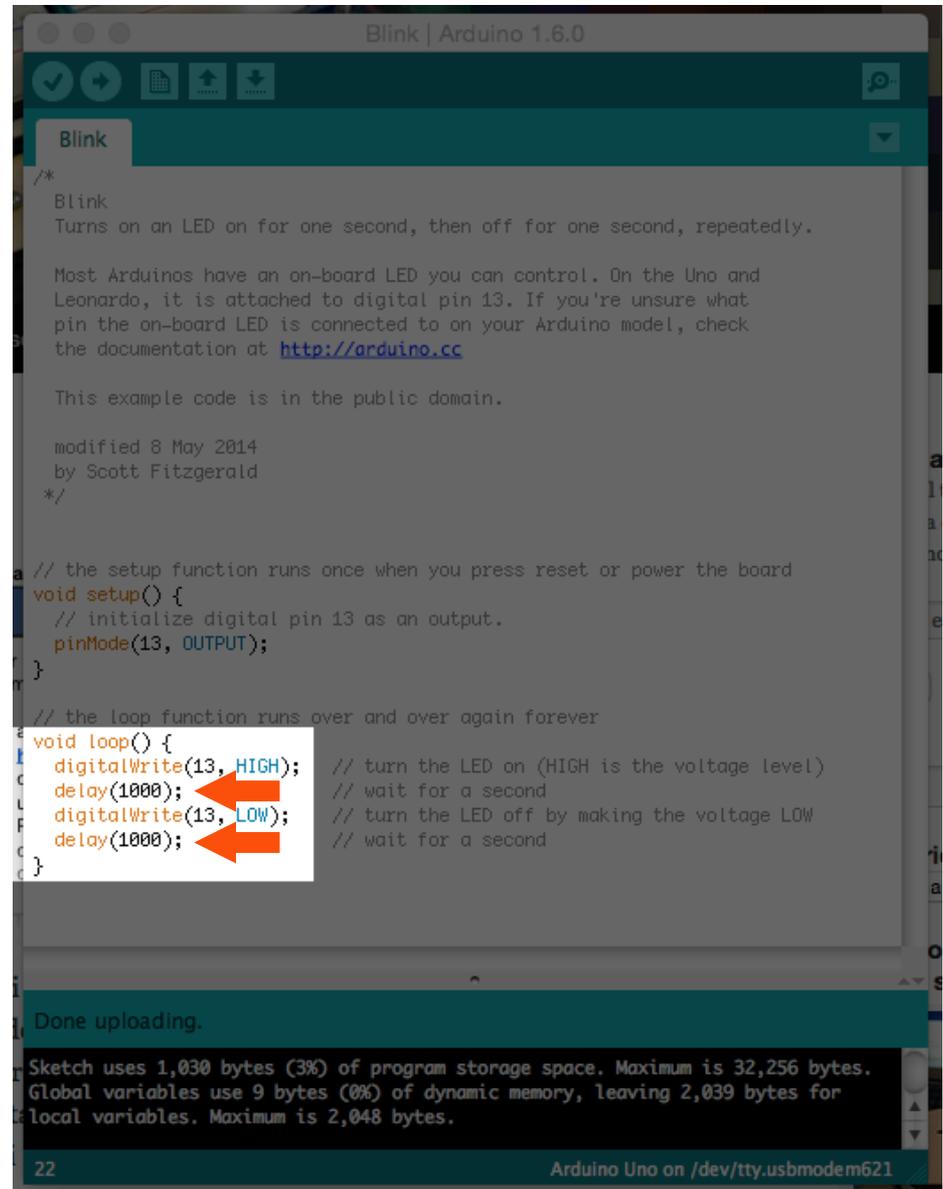
Done uploading.

Sketch uses 1,030 bytes (3%) of program storage space. Maximum is 32,256 bytes.  
Global variables use 9 bytes (0%) of dynamic memory, leaving 2,039 bytes for local variables. Maximum is 2,048 bytes.

22 Arduino Uno on /dev/tty.usbmodem621

Sulla scheda Arduino, se nuova e mai utilizzata, viene precaricato lo sketch Blink, quindi appena viene collegata la scheda al computer il LED L lampeggia.

Per essere certi che lo sketch è stato caricato sulla scheda provate a variare il numero all'interno del comando delay, ponete il valore 100. Effettuate l'upload dello sketch, al termine dovrete notare che il LED L lampeggia molto più velocemente.



```
Blink | Arduino 1.6.0
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
  */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

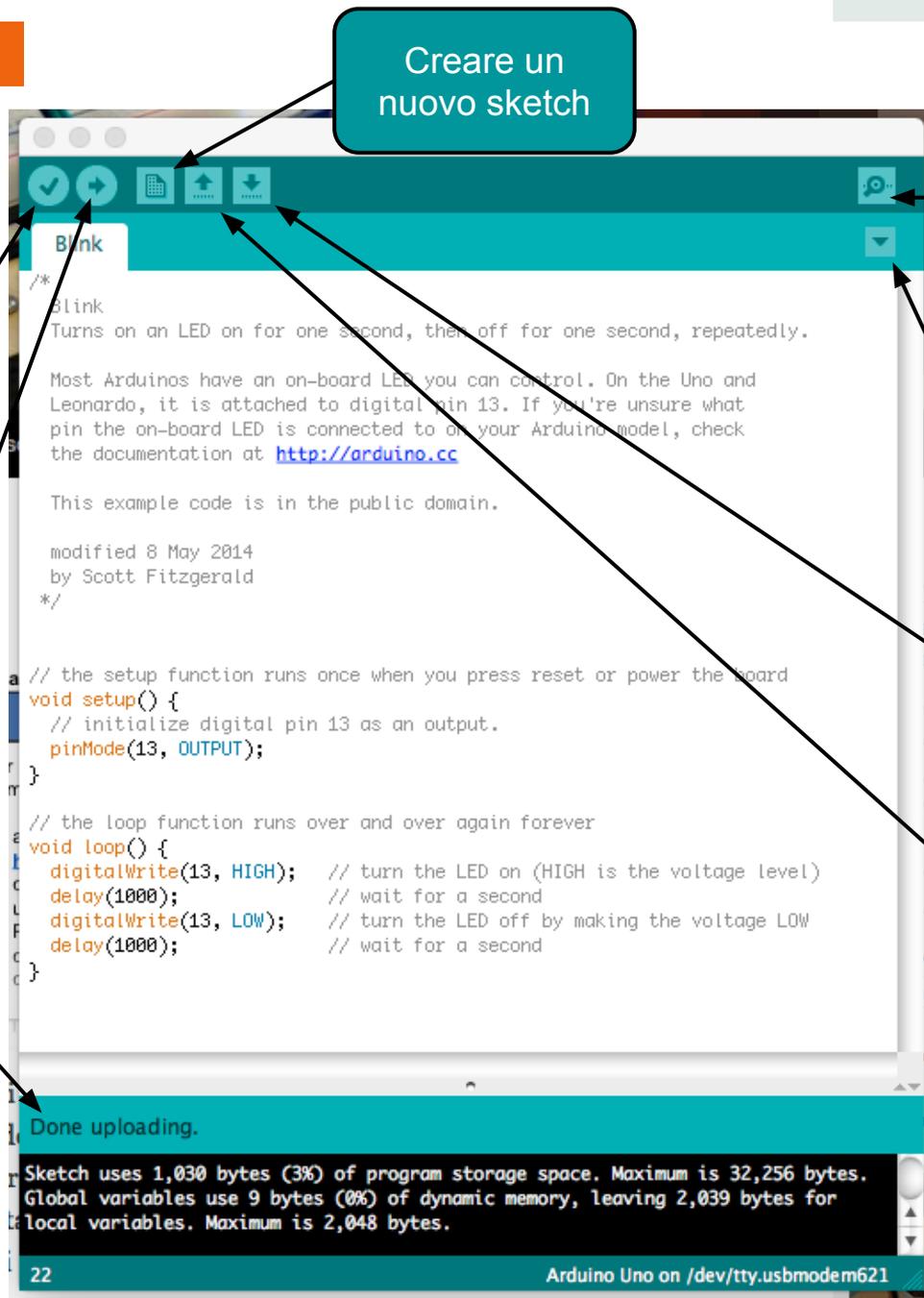
Done uploading.

Sketch uses 1,030 bytes (3%) of program storage space. Maximum is 32,256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2,039 bytes for
local variables. Maximum is 2,048 bytes.

22 Arduino Uno on /dev/tty.usbmodem621
```

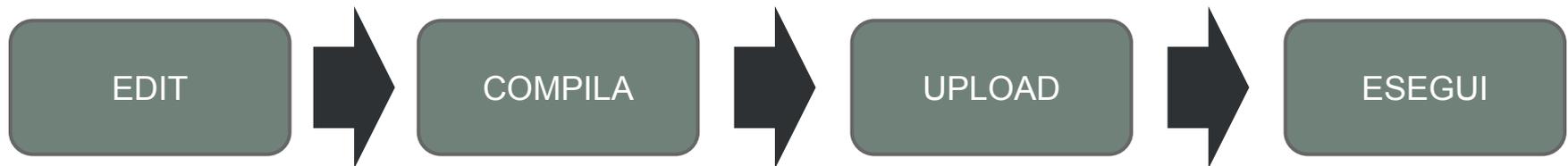
# Programmazione

1' IDE



## Ciclo di sviluppo

Il ciclo di sviluppo è suddiviso in 4 fasi:



**Compila.** *Compilare vuol dire tradurre lo sketch in linguaggio macchina, detto anche codice oggetto*  
**Esegui.** *uno sketch Arduino viene eseguito non appena termina la fase di upload sulla scheda*

## Il linguaggio

Il linguaggio di programmazione è un C standard (ma molto più semplice)

Le funzioni più usate e che impareremo ad utilizzare durante le lezioni sono:

**pinMode()**

impostare un pin come input o come output

**digitalWrite()**

impostare un pin digitale a HIGH o LOW

**digitalRead()**

legge lo stato di un pin digitale

**analogRead()**

legge un pin analogico

**analogWrite()**

scrive in valore analogico

**delay()**

mette in attesa il programma per un determinato tempo

**millis()**

restituisce l'ora corrente (tempo di accensione di Arduino)

Altre funzioni con esempi di utilizzo potete trovarle seguendo il [link](#).

# **Il primo programma**

```
/*  
Blink  
Turns on an LED on for one second, then off for one second, repeated  
  
Most Arduinos have an on-board LED you can control. On the Uno and  
Leonardo, it is attached to digital pin 13. If you're unsure what  
pin the on-board LED is connected to on your Arduino model, check  
the documentation at http://arduino.cc  
  
This example code is in the public domain.  
  
modified 8 May 2014  
by Scott Fitzgerald  
*/
```

A

Commento su più linee

```
B → // the setup function runs once when you press reset or power the board  
void setup() {  
B → // initialize digital pin 13 as an output.  
  pinMode(13, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level) ← B  
  delay(1000); // wait for a second ← B  
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW ← B  
  delay(1000); // wait for a second ← B  
}
```

B

Commento su una linea

```
/*  
Blink  
Turns on an LED on for one second, then off for one second, repeatedly.
```

```
Most Arduinos have an on-board LED you can control. On the  
Leonardo, it is attached to digital pin 13. On other boards, you  
may need to use a pin that is marked as a LED pin, for example  
pin the on-board LED is connected to on the Uno Rev3. See  
the documentation at http://arduino.cc
```

```
This example code is in the public domain.
```

```
modified 8 May 2014
```

```
by Scott Fitzgerald
```

```
*/
```

```
// the setup function runs once when you power up the board  
void setup() {
```

```
  // initialize digital pin 13 as an output
```

```
  pinMode(13, OUTPUT);  
}
```

```
// the loop function runs over and over again
```

```
void loop() {
```

```
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
```

```
  delay(1000); // wait for a second
```

```
  digitalWrite(13, LOW); // turn the LED off by making the pin LOW
```

```
  delay(1000); // wait for a second
```

```
}
```

A white semicolon symbol is centered on a red square background.

identifica dove  
termina  
un'istruzione

A white opening curly brace is centered on a red square background.Three white dots are arranged horizontally in the center of a red square background.A white closing curly brace is centered on a red square background.

identifica un blocco  
di istruzioni

```
// the setup function runs once when you press reset or power
```

```
void setup() {  
  // initialize digital pin 13 as an output.  
  pinMode(13, OUTPUT);  
}
```

```
// the loop function runs over and over again forever
```

```
void loop() {  
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the positive voltage)  
  delay(1000);              // wait for a second  
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW  
  delay(1000);              // wait for a second  
}
```

## Struttura di base

```
void setup()  
{  
  istruzioni;  
}
```

```
void loop()  
{  
  istruzioni;  
}
```

La struttura base di un programma Arduino è abbastanza semplice e si sviluppa in almeno due parti. Queste due parti, o funzioni, necessarie racchiudono parti di istruzioni.

Dove **setup()** indica il blocco di settaggio e **loop()** è il blocco che viene eseguito. Entrambe le sezioni sono necessarie per far sì che uno sketch funzioni.

**setup()** è la prima funzione ad essere invocata verrà eseguita una volta sola e in essa vengono dichiarate le variabili usate nel programma, è usata per impostare il pinMode o inizializzare la comunicazione seriale.

La funzione **loop()** contiene il codice che deve essere eseguito ripetutamente, in essa vengono letti gli input, i segnali di output ecc...

Questa funzione è la parte principale di un programma Arduino (sketch), esegue la maggior parte del lavoro.

```
void setup()
{
  istruzioni;
}

void loop()
{
  istruzioni;
}
```

*Per approfondimenti seguire il [link](#).*

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  Most Arduinos have an on-board LED you can control. On the Uno and  
  Leonardo, it is attached to digital pin 13. On the Mega and  
  pin the on-board LED is connected to on-board pin 11. Check  
  the documentation at http://arduino.cc  
  for more information.  
  
  This example code is in the public domain.  
  
  modified 8 May 2014  
  by Scott Fitzgerald  
*/  
  
// the setup function runs once when you power up the board  
void setup() {  
  // initialize digital pin 13 as an output  
  pinMode(13, OUTPUT);  
}  
  
// the loop function runs over and over again  
void loop() {  
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW  
  delay(1000);           // wait for a second  
}
```

## `pinMode(n, OUTPUT);`

`pinMode` è un'istruzione che dice ad Arduino come usare un determinato pin.

Tra parentesi tonde vengono specificati gli argomenti che possono essere numeri e lettere.

I pin digitali possono essere utilizzati sia come **INPUT** che come **OUTPUT**.

Nel nostro caso poiché vogliamo far lampeggiare il diodo LED dobbiamo definire il pin di **OUTPUT**.

Le parole **INPUT** e **OUTPUT** sono costanti definite, che non variano mai nel linguaggio di Arduino.

**Per approfondimenti seguire il [link](#).**



```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second  
  
  Most Arduinos have an on-board LED you can control. On the  
  Leonardo, it is attached to digital pin 13. On the Uno and  
  other boards, pin the on-board LED is connected to on your  
  the documentation at http://arduino.cc  
  
  This example code is in the public domain.  
  
  modified 8 May 2014  
  by Scott Fitzgerald  
*/  
  
// the setup function runs once when you press reset  
void setup() {  
  // initialize digital pin 13 as an output.  
  pinMode(13, OUTPUT);  
}  
  
// the loop function runs over and over again  
void loop() {  
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);            // wait for a second  
  digitalWrite(13, LOW);  // turn the LED off by making the pin LOW  
  delay(1000);            // wait for a second  
}
```

## digitalWrite(n, HIGH);

L'istruzione digitalWrite possiede due argomenti:

il primo definisce il pin,  
il secondo indica lo stato.

digitalWrite è un'istruzione in grado di impostare un pin definito come OUTPUT ad un valore HIGH o ad un valore LOW, in modo più semplice permette di accendere o spegnere un led connesso al pin specificato nel primo argomento, nel nostro caso LED.

Il 'pin' può essere specificato come una variabile o una costante (0-13).

Tenete conto che dire che su un determinato pin vi è uno stato HIGH, vuol dire che su di esso viene applicata una tensione di +5 V, mentre se lo stato è LOW vuol dire che sul pin è applicata una tensione di 0V.

**Per approfondimenti seguire il [link](#).**

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  Most Arduinos have an on-board LED you can control. On the Uno and  
  Leonardo, it is attached to digital pin 13. If you're unsure what  
  pin the on-board LED is connected to on your Arduino model, check  
  the documentation at http://arduino.cc  
  
  This example code is in the public domain.  
  
  modified 8 May 2014  
  by Scott Fitzgerald  
*/  
  
// the setup function runs once when you press reset  
void setup() {  
  // initialize digital pin 13 as an output.  
  pinMode(13, OUTPUT);  
}  
  
// the loop function runs over and over again  
void loop() {  
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);            // wait for a second  
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW  
  delay(1000);           // wait for a second  
}
```

## **delay(1000);**

delay() è un'istruzione che interrompe per un determinato tempo l'esecuzione del programma.

L'istruzione ha un solo argomento numerico che indica il numero di **millisecondi** di attesa.

Con “**delay(1000)**” il programma si bloccherà per 1000 millisecondi ovvero 1 secondo.

**Per approfondimenti seguire il [link](#).**

# **Prodotti Arduino**

Board



Arduino Uno



Arduino Leonardo



Arduino Micro



Arduino Esplora



Arduino Due



Arduino Yún



Arduino Mega ADK



Arduino Ethernet



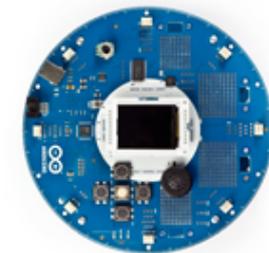
Arduino Tre



Arduino Zero



Arduino Mega 2560



Arduino Robot

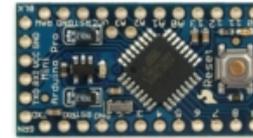
Board



Arduino Mini



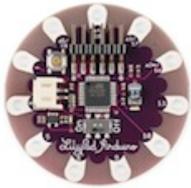
Arduino Nano



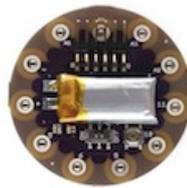
Arduino Pro Mini



Arduino Fio



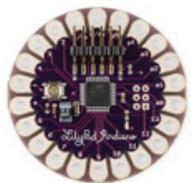
LilyPad Arduino Simple



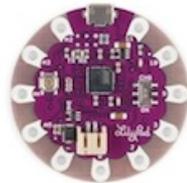
LilyPad Arduino SimpleSnap



Arduino Pro



LilyPad Arduino



LilyPad Arduino USB

Shield



Arduino GSM Shield



Arduino Wireless SD  
Shield



Arduino Motor Shield



Arduino Ethernet  
Shield



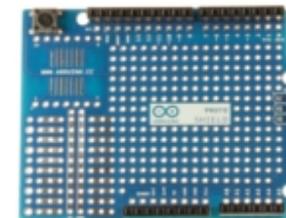
Arduino USB Host  
Shield



Arduino Wireless Proto  
Shield



Arduino WiFi Shield



Arduino Proto Shield

Kit e Accessori

KIT



The Arduino Starter Kit



Arduino Materia 101

ACCESSORI



TFT LCD screen



Arduino ISP



USB/Serial Light Adapter



Mini USB/Serial Adapter

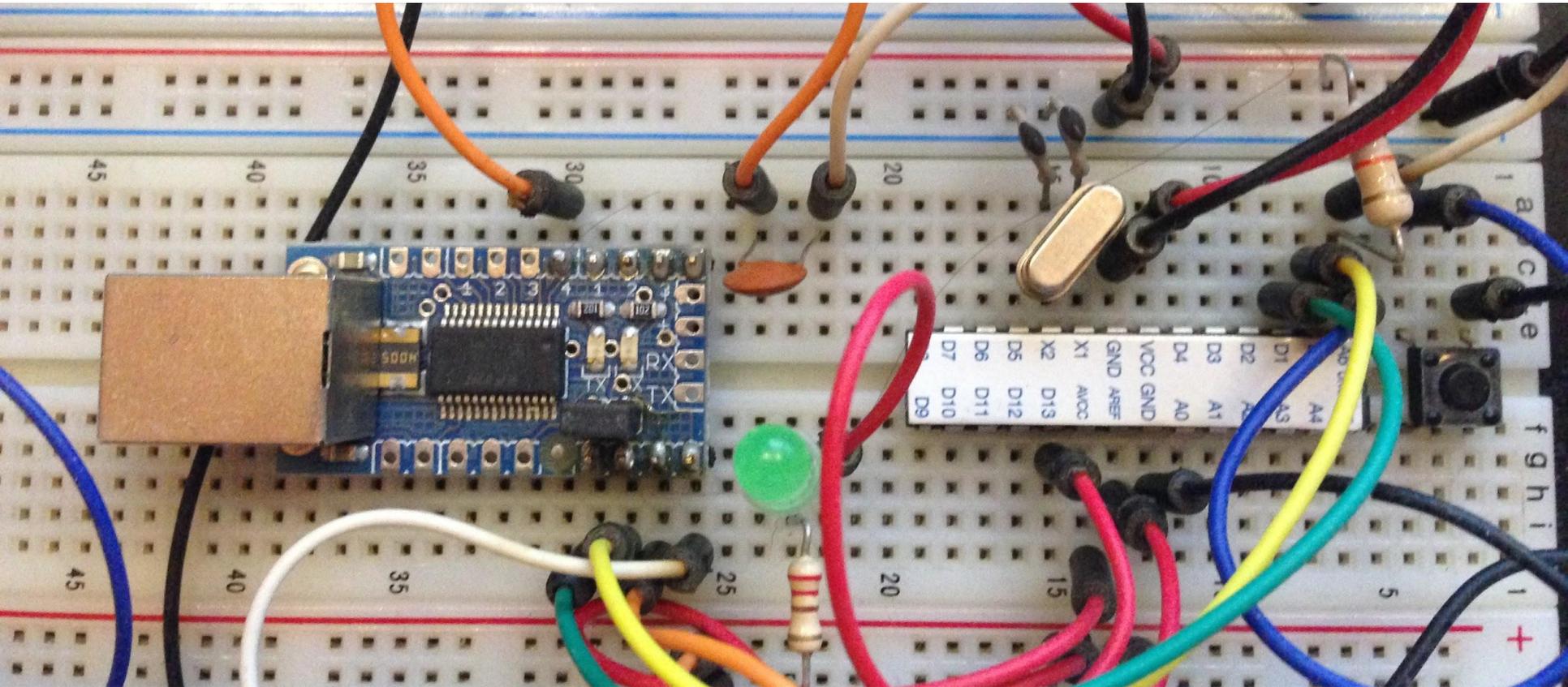
**Ma quale scheda  
scegliere per il mio  
progetto?**

# Compare board specs

This simple table shows a quick comparison between the characteristics of all the Arduino boards.

Name	Processor	Operating Voltage/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [KB]	SRAM [KB]	Flash [KB]	USB	UART
Uno	ATmega328	5 V/7-12 V	16MHz	6/0	14/6	1	2	32	Regular	1
Due	AT91SAM3X8E	3.3 V/7-12 V	84 MHz	12/2	54/12	-	96	512	2 Micro	4
Leonardo	ATmega32u4	5 V/7-12 V	16MHz	12/0	20/7	1	2.5	32	Micro	1
Mega 2560	ATmega2560	5 V/7-12 V	16MHz	16/0	54/15	4	8	256	Regular	4
Mega ADK	ATmega2560	5 V/7-12 V	16MHz	16/0	54/15	4	8	256	Regular	4
Micro	ATmega32u4	5 V/7-12 V	16MHz	12/0	20/7	1	2.5	32	Micro	1
Mini	ATmega328	5 V/7-9 V	16MHz	8/0	14/6	1	2	32	-	-
Nano	ATmega168	5 V/7-9 V	16MHz	8/0	14/6	0.512	1	16	Mini-B	1
	ATmega328					1	2	32		
Ethernet	ATmega328	5 V/7-12 V	16MHz	6/0	14/4	1	2	32	Regular	-
Esplora	ATmega32u4	5 V/7-12 V	16MHz	-	-	1	2.5	32	Micro	-
ArduinoBT	ATmega328	5 V/2.5-12 V	16MHz	6/0	14/6	1	2	32	-	1
Fio	ATmega328P	3.3 V/3.7-7 V	8MHz	8/0	14/6	1	2	32	Mini	1

# **Costruirsi una scheda Arduino**

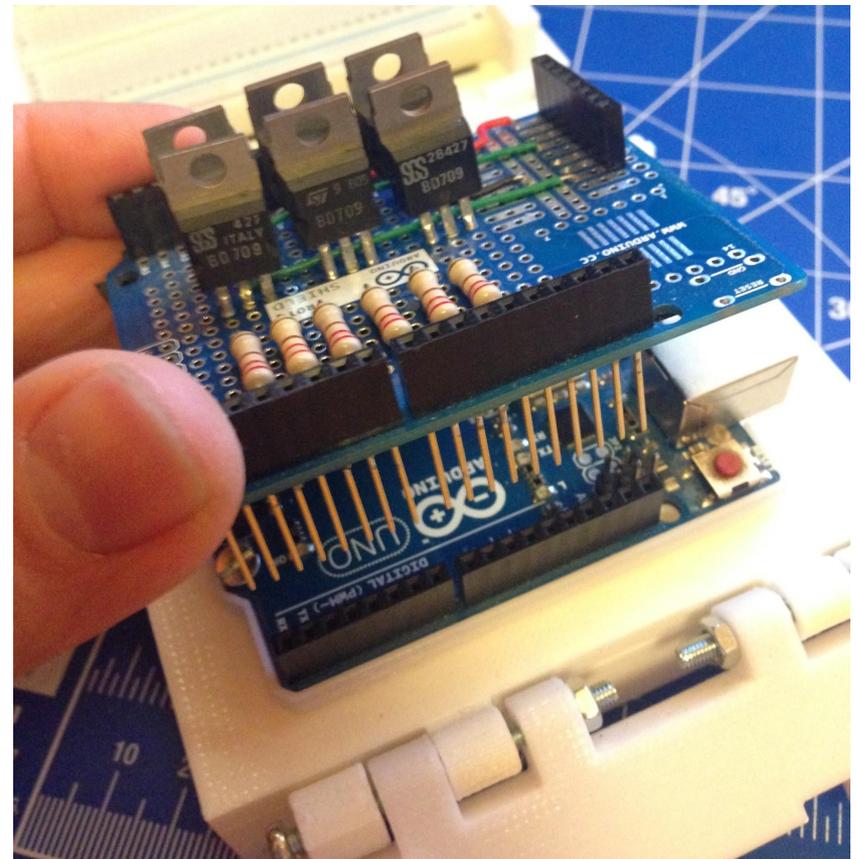
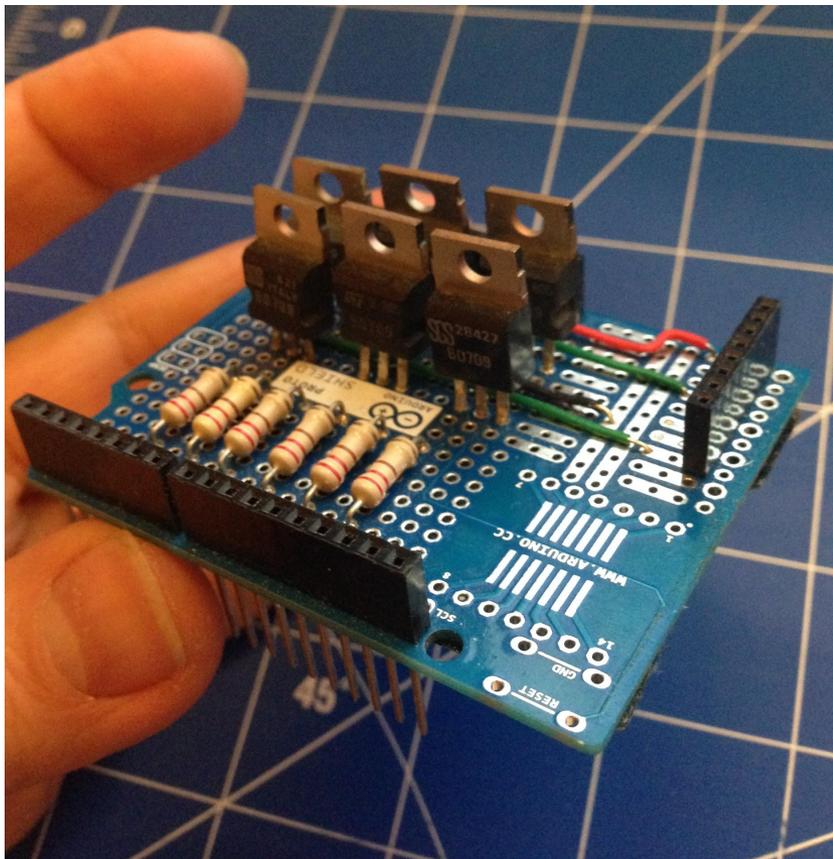


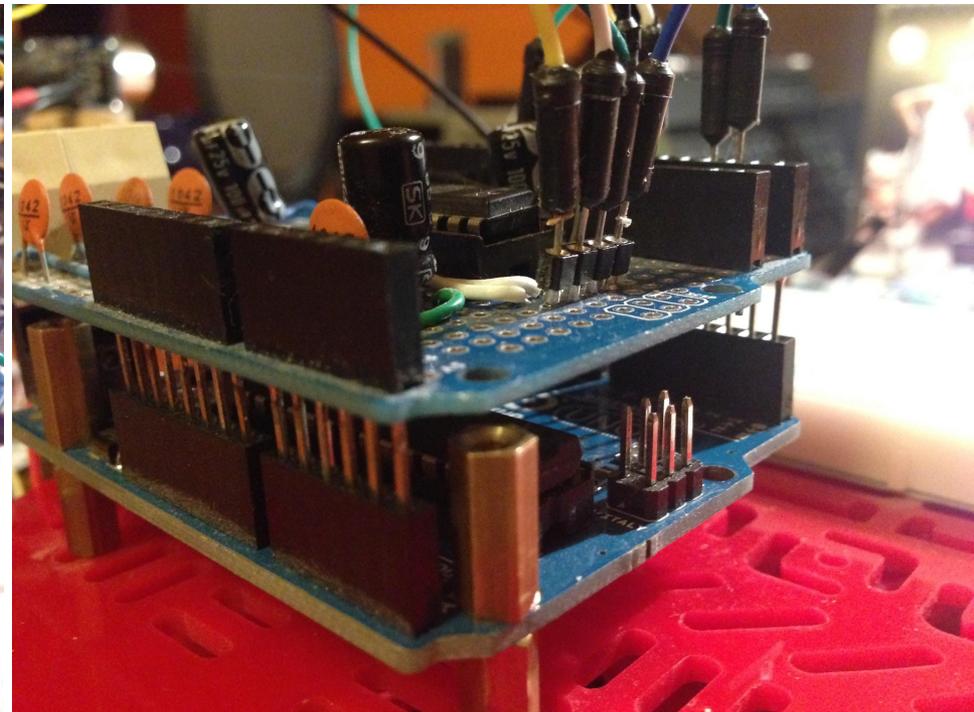
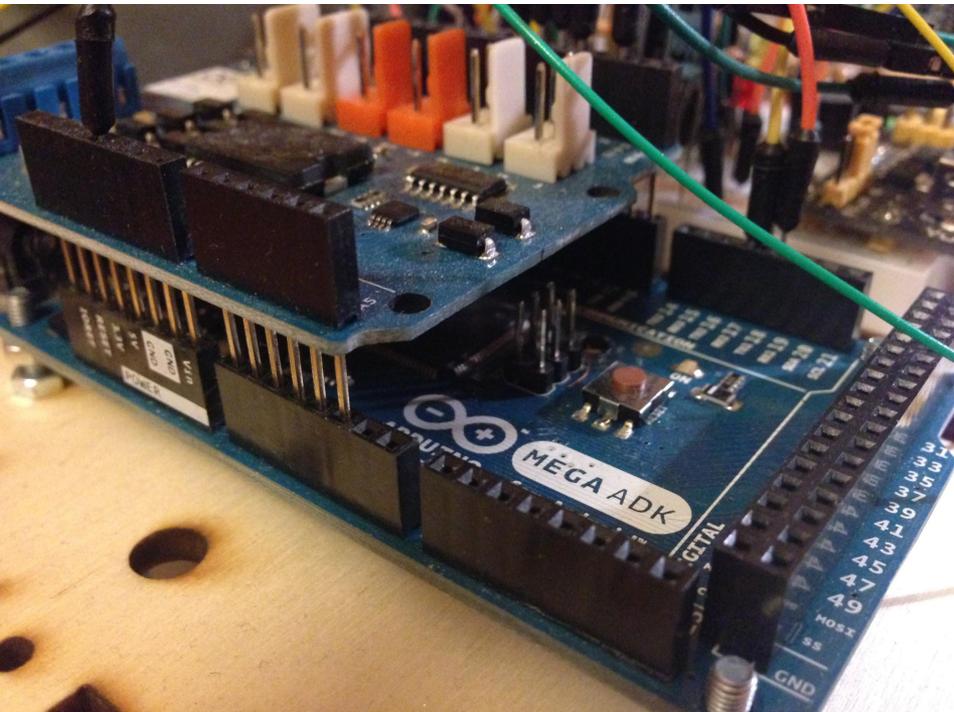
# **Estendere le funzionalità con gli Shield**

## Shield

1/3

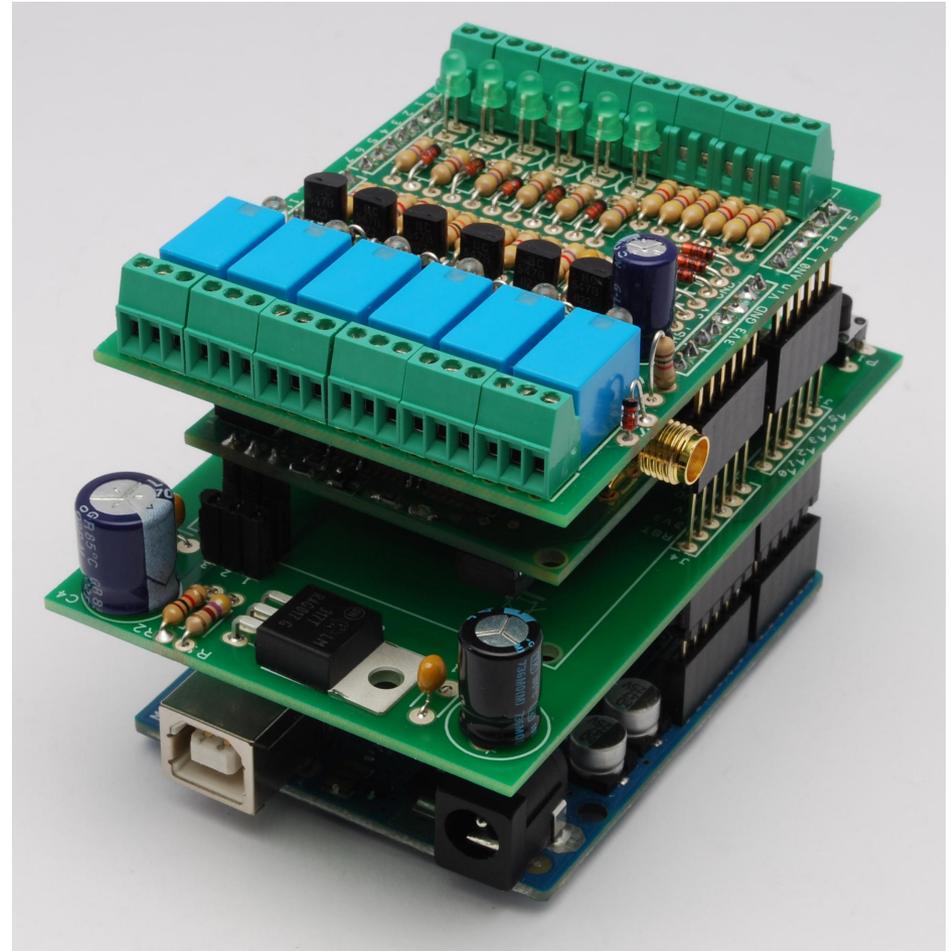
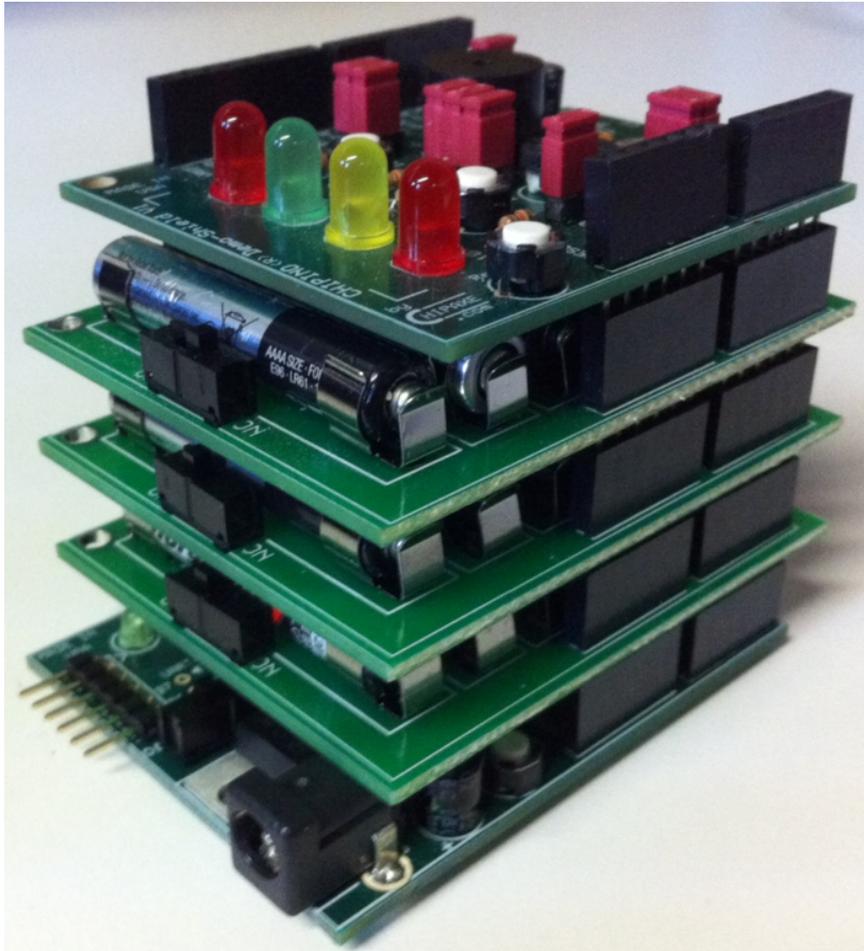
Gli shield sono schede elettroniche che si collegano (impilano) ad Arduino e permettono di aggiungere funzionalità aggiuntive aumentando il livello di interazione con il mondo esterno. In commercio esiste una vastissima gamma di shield, ma l'utente è libero di realizzarne di personali, saldando i componenti su basette millefori predisposte o realizzando pcb ad hoc.





Shield

3/3



## Pensieri

L'insuccesso *nel proprio "programma di apprendimento"* accade perché non vengono incluse le librerie (funzioni operative) necessarie per affrontare risolvere i problemi.

```
#include (meraviglia.h)
#include (precisionePignoleria.h)
#include (umilta.h)
#include (openSorce.h)
#include (dream.h)
```

in altro modo:

guardare al mondo con meraviglia

essere precisi nelle proprie azioni di progetto

tanta umiltà

pensiero ed azioni didattiche open

sognare

# Grazie

Prof. Michele Maffucci

[www.maffucci.it](http://www.maffucci.it)

michele@maffucci.it

[www.twitter.com/maffucci/](http://www.twitter.com/maffucci/)

[www.facebook.com/maffucci.it/](http://www.facebook.com/maffucci.it/)

[plus.google.com/+MicheleMaffucci/](https://plus.google.com/+MicheleMaffucci/)

[it.linkedin.com/in/maffucci](http://it.linkedin.com/in/maffucci)

Licenza presentazione:

