



OPEN SOURCE-

"RISORSE SOFTWARE O HARDWARE CHE POSSONO ESSERE UTILIZZATE, RIDISTRIBUITE O MODIFICATE GRATUITAMENTE."

ELETTRONICA-

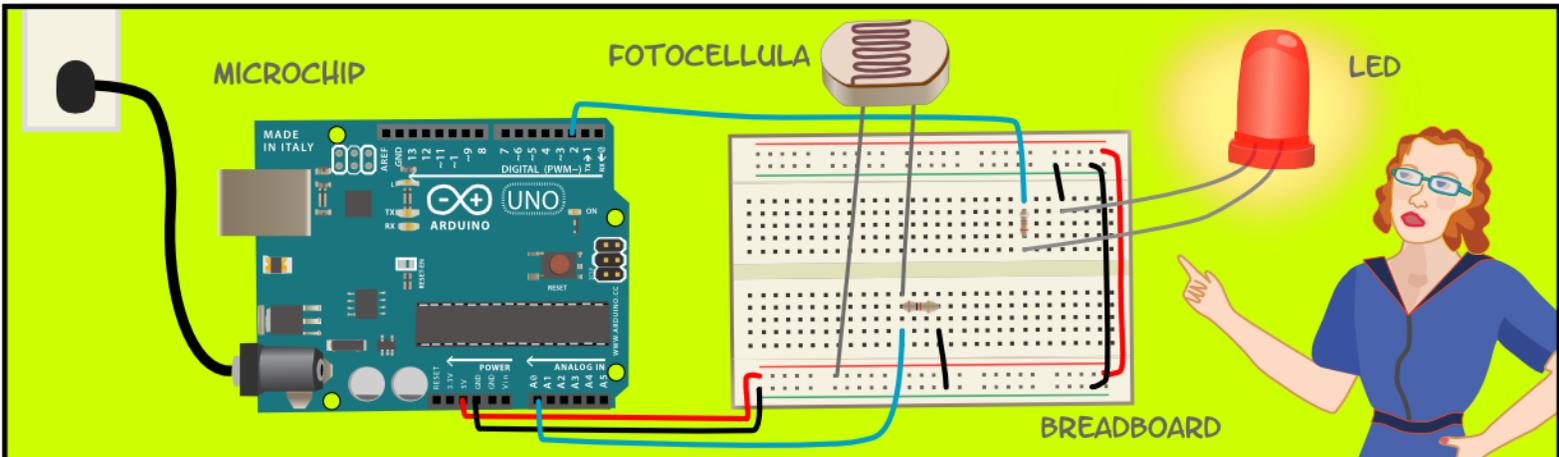
"TECNOLOGIA CHE SFRUTTA IL MOTO CONTROLLATO DEGLI ELETTRONI ATTRAVERSO I MATERIALI"

PROTOTIPO-

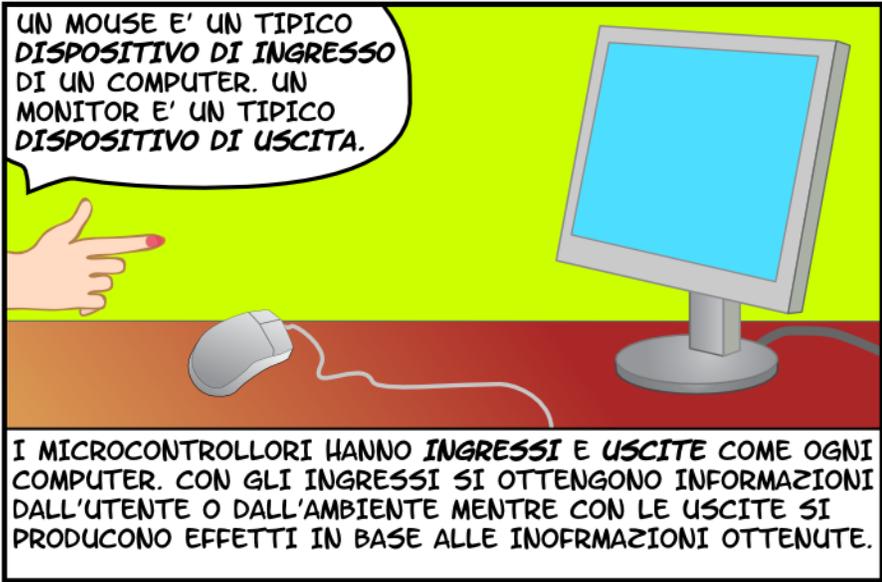
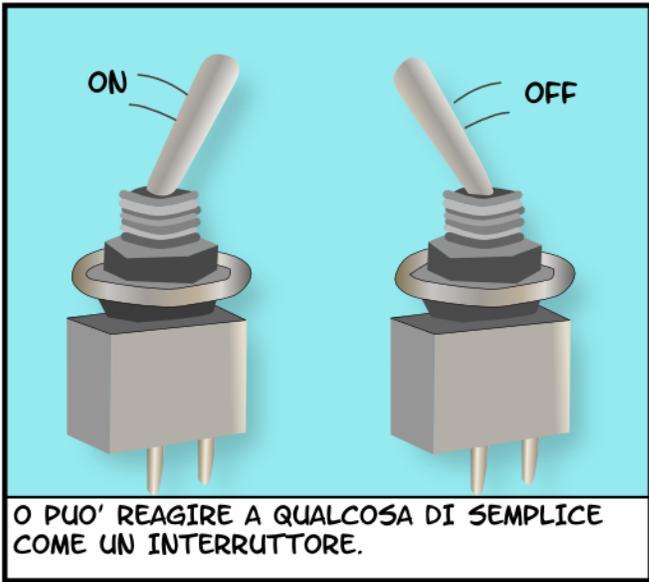
"UN PROGETTO ORIGINALE CHE PUO' SERVIRE COME BASE PER ALTRI OGGETTI"

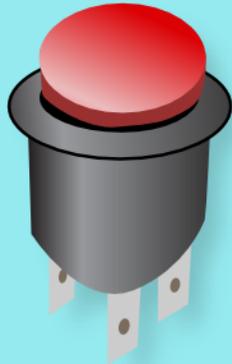
PIATTAFORMA-

"HARDWARE CON SOFTWARE DI SUPPORTO SU CUI POSSONO ESSERE ESEGUITI DEI PROGRAMMI"

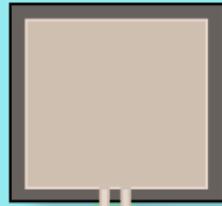


ARDUINO CONTIENE UN **MICROCHIP** CHE E' UN PICCOLO COMPUTER PROGRAMMABILE (IL MICROCONTROLLORE). ALLA SCHEDA E' POSSIBILE COLLEGARE DEI SENSORI CHE MISURANO LE CONDIZIONI AMBIENTALI (AD ESEMPIO QUANTA LUCE C'E' IN UNA STANZA) PER CONTROLLARE IL MODO IN CUI ALTRI OGGETTI REAGISCONO A QUESTE CONDIZIONI (SE LA STANZA DIVENTA BUIA UN LED SI ACCENDE).





PULSANTE



SENSORE
DI FORZA
RESISTIVO

UN PULSANTE O UN SENSORE POSSONO ESSERE USATI COME INGRESSI.



MOTORE
IN CONTINUA



OGNI OGGETTO CHE PUO' ESSERE ACCESO, SPENTO E CONTROLLATO PUO' ESSERE UN'USCITA. AD ESEMPIO UN MOTORE MA ANCHE UN COMPUTER.



QUAL E' LA DIFFERENZA
TRA INGRESSI E USCITE
DIGITALI ED
ANALOGICI?

GLI **INGRESSI** E LE **USCITE** POSSONO ESSERE DIGITALI O ANALOGICI. UN'INFORMAZIONE DIGITALE E' BINARIA (E' VERA OPPURE FALSA). UNA ANALOGICA E' CONTINUA E PUO' ASSUMERE MOLTI VALORI.

LE **INFORMAZIONI DIGITALI** SONO **DISCRETE** E **FINITE**. OGNI INFORMAZIONE E' DESCRITTA CON **DUE STATI**, **1 O 0**, **ACCESSO** O **SPENTO**.

LE **INFORMAZIONI ANALOGICHE** SONO **CONTINUE** E POSSONO ASSUMERE UN **NUMERO INFINITO** DI VALORI.



UN INTERRUPTORE E' UN **INGRESSO DIGITALE**. UN SENSORE E' UN **INGRESSO ANALOGICO** I CUI VALORI DIPENDONO DALLA **CONVERSIONE DIGITALE** DEI DATI.

TENSIONE?
CORRENTE?
RESISTENZA?
LEGGE DI OHM?



PER USARE ARDUINO BISOGNA CAPIRE COME FUNZIONA L'ELETTRICITA' (E QUINDI L'ELETTRONICA). VEDIAMO QUALCHE CONCETTO FONDAMENTALE.

LA **TENSIONE (V)**
E' UNA MISURA
DEL POTENZIALE
ELETTRICO.
SI MISURA
IN **VOLT**.

LA **CORRENTE (I)**
E' IL FLUSSO
DI CARICA IN
UN CONDUTTORE.
SI MISURA
IN **AMPERE**.

LA **RESISTENZA (R)**
DICE QUANTO
UN CONDUTTORE
SI OPPONE AL
PASSAGGIO DI
CORRENTE. SI
MISURA IN **OHM**.

L'ELETTRICITA' E' IL MOVIMENTO DI CARICHE IN UN CONDUTTORE.

LA VELOCITA' DEL FLUSSO
DIPENDE DALLA **TENSIONE**

LA **RESISTENZA** AUMENTA
O DIMINUISCE IL FLUSSO

IL FLUSSO NEL TUBO
E' LA **CORRENTE**

SPESSO PER SPIEGARE QUESTI TERMINI SI RICORRE ALL'ANALOGIA CON L'ACQUA, COME IN QUESTO MODELLO.

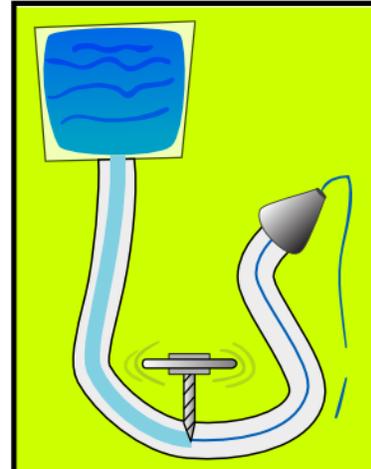
LEGGE DI OHM

CORRENTE = TENSIONE/RESISTENZA
($I = V/R$)

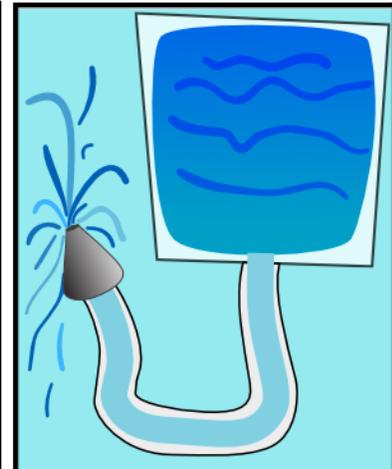
OPPURE
RESISTENZA = TENSIONE/CORRENTE
($R = V/I$)

OPPURE
TENSIONE = RESISTENZA * CORRENTE
($V = R*I$)

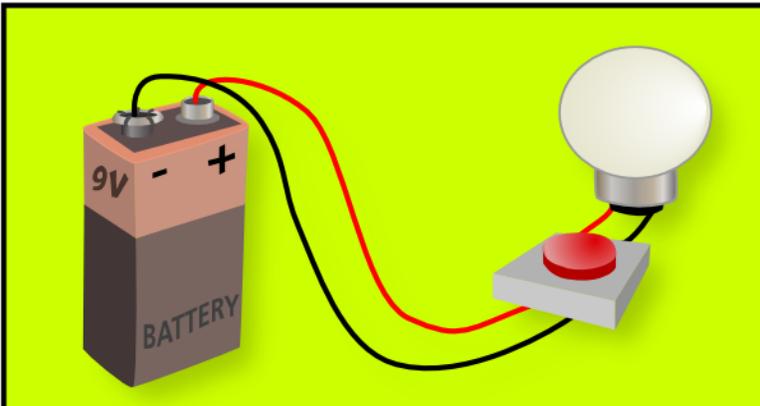
ESISTE UNA RELAZIONE TRA TENSIONE, CORRENTE E RESISTENZA SCOPERTA DA GEORG OHM, UN FISICO TEDESCO.



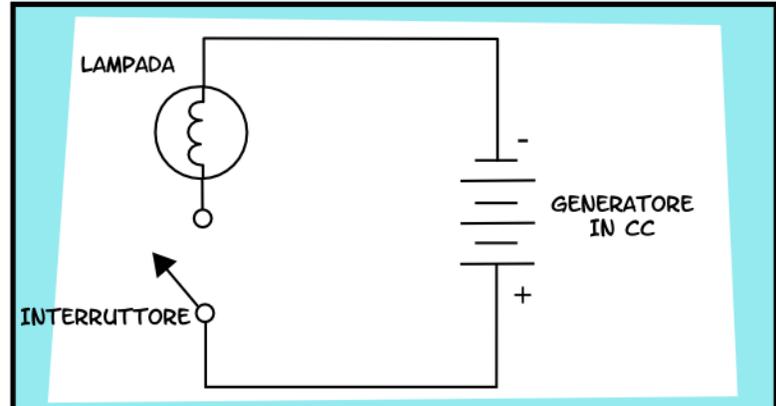
PER ESEMPIO AUMENTANDO LA RESISTENZA DIMINUISCE IL FLUSSO.



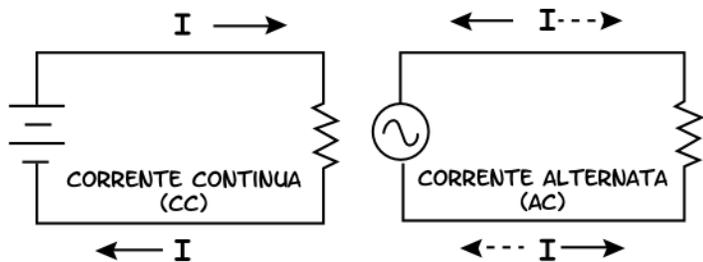
IL FLUSSO AUMENTA SE CRESCE IL POTENZIALE.



VEDIAMO UN SEMPLICE CIRCUITO. UN CIRCUITO E' UN PERCORSO CHIUSO CON UNA SORGENTE DI ENERGIA (BATTERIA) E UN CARICO (LAMPADA). IL CARICO UTILIZZA L'ENERGIA ELETTRICA DELLA BATTERIA E LA TRASFORMA. NEL CIRCUITO C'E' UN INTERRUTTORE.



QUESTO E' UNO SCHEMA ELETTRICO DELLO STESSO CIRCUITO DOVE I COMPONENTI SONO RAPPRESENTATI CON DEI SIMBOLI. QUANDO L'INTERRUTTORE E' CHIUSO LA CORRENTE CIRCOLA E FA ACCENDERE LA LAMPADA.



ESISTONO DUE TIPI DI CIRCUITI: QUELLI IN **CORRENTE CONTINUA** E QUELLI IN **CORRENTE ALTERNATA**. NEI CIRCUITI IN CC LA CORRENTE VA IN UN SOLO VERSO. IN QUELLI IN ALTERNATA IL VERSO CAMBIA CICLICAMENTE. A NOI INTERESSANO SOLO I CIRCUITI IN CC.

ORA CHE ABBIAMO VISTO COME FUNZIONA L'ELETTRICITA' TORNIAMO AD ARDUINO.



ARDUINO HA BISOGNO DI ENERGIA PER FUNZIONARE E DI UN COMPUTER PER PROGRAMMARLO.



COLLEGANDO ARDUINO A UN COMPUTER CON UN CAVO USB FORNIAMO L'ALIMENTAZIONE E POSSIAMO COMINCIARE A PROGRAMMARLO.



SCARCA QUI IL SOFTWARE:

[HTTP://ARDUINO.CC/EN/MAIN/SOFTWARE](http://arduino.cc/en/main/software)

PER PROGRAMMARE ARDUINO BISOGNA SCARICARE E INSTALLARE DEL SOFTWARE. IL SOFTWARE, CHE E' GRATUITO E DISPONIBILE ALL'INDIRIZZO INDICATO, GIRA SULLE PIATTAFORME MAC OS X, WINDOWS E LINUX.

ISTRUZIONI PER INSTALLARE IL SOFTWARE ARDUINO SU MAC:

[HTTP://WWW.ARDUINO.CC/EN/GUIDE/MACOSX](http://www.arduino.cc/en/guide/macOSX)

ISTRUZIONI PER INSTALLARE IL SOFTWARE ARDUINO SU WINDOWS:

[HTTP://WWW.ARDUINO.CC/EN/GUIDE/WINDOWS](http://www.arduino.cc/en/guide/windows)

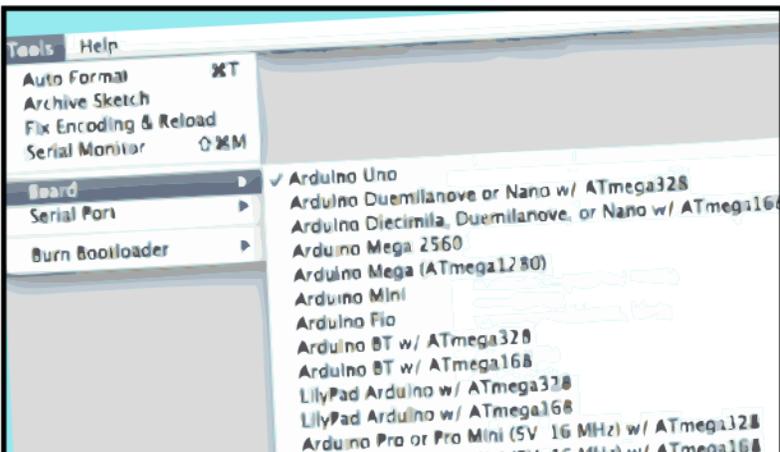
ISTRUZIONI PER INSTALLARE IL SOFTWARE ARDUINO SU LINUX:

[HTTP://WWW.ARDUINO.CC/EN/GUIDE/LINUX](http://www.arduino.cc/en/guide/linux)

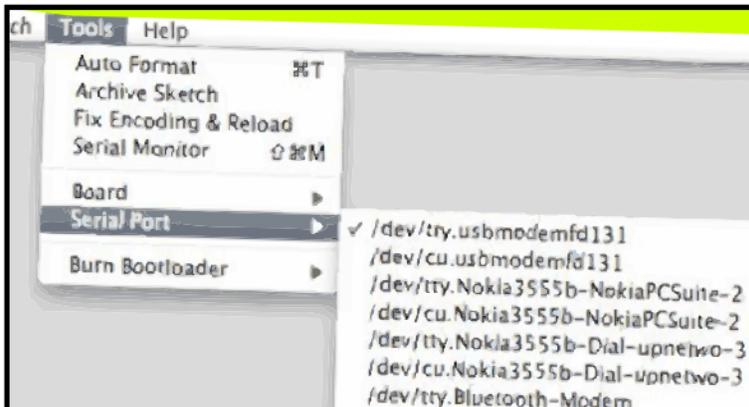
LE ISTRUZIONI DETTAGLIATE PER INSTALLARE IL SOFTWARE SULLE VARIE PIATTAFORME SI TROVANO A QUESTI INDIRIZZI.



QUANDO HAI INSTALLATO IL SOFTWARE COLLEGA ARDUINO. IL LED ON SULLA SCHEDA DOVREBBE ACCENDERSI.



LANCIA IL SOFTWARE ARDUINO. NEL MENU TOOLS SCEGLI LA SCHEDA CHE UTILIZZI (TOOLS > BOARD). PER ESEMPIO ARDUINO UNO.

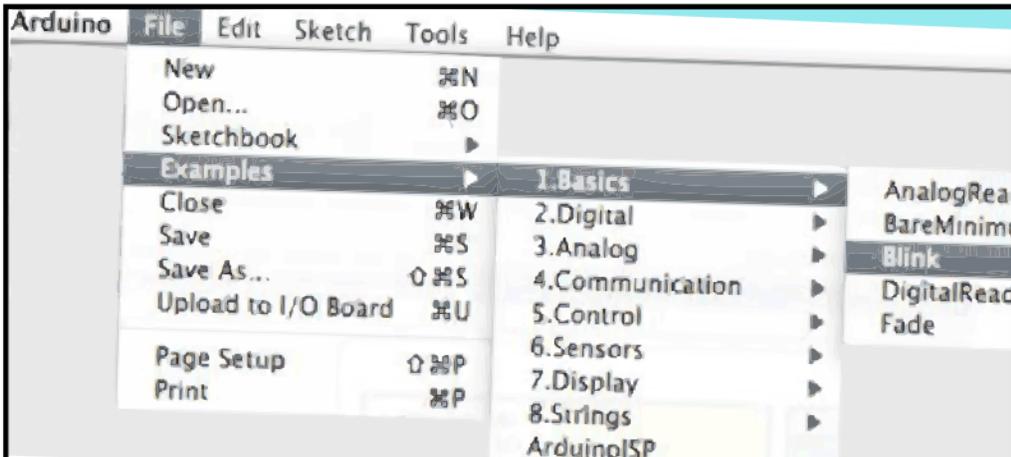


ORA SCEGLI LA PORTA SERIALE (TOOLS > SERIAL PORT). IN UN MAC POTREBBE ESSERE /DEV/TTY.USBMODEM. IN WINDOWS SARA' COM3 O QUALCOSA DI SIMILE.

COS'E' UN
INTEGRATED
DEVELOPMENT
ENVIRONMENT?



IL SOFTWARE ARDUINO E' UN IDE (AMBIENTE DI SVILUPPO INTEGRATO). COMBINA UN EDITOR DI TESTO CON UN COMPILATORE E ALTRE UTILITA' E SERVE AI PROGRAMMATORI PER SVILUPPARE DEL SOFTWARE.



L'IDE ARDUINO SERVE A SCRIVERE I PROGRAMMI, DETTI SKETCH, E A CARICARLI SULLA SCHEDA ARDUINO. APRI L'ESEMPIO BLINK DAL MENU FILE. FILE > EXAMPLES > 1.BASICS > BLINK.



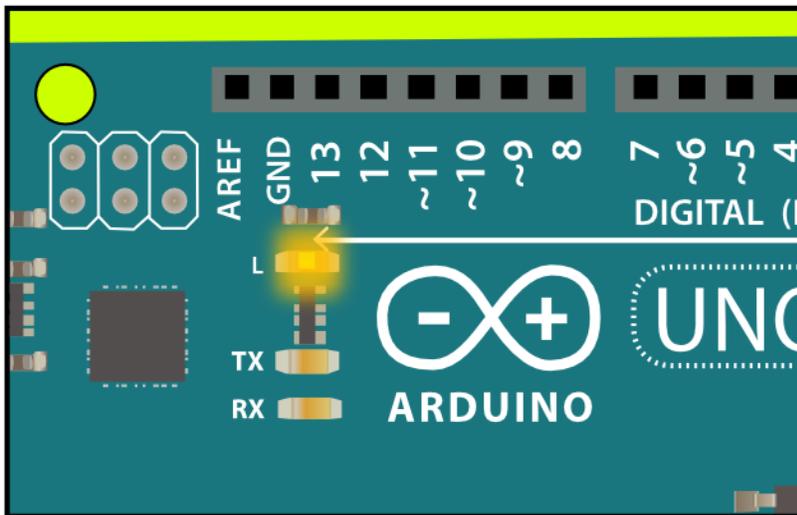
BOTTONE UPLOAD

```
int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
```

PER CARICARE UNO SKETCH SULLA SCHEDA CLICCA SUL BOTTONE UPLOAD NELLA BARRA DEGLI STRUMENTI IN CIMA ALLA FINESTRA. NELLA PARTE BASSA DELLA FINESTRA COMPARIRANNO DEI MESSAGGI E ALLA FINE DONE UPLOADING.



IL LED COLLEGATO AL PIN 13 DELLA SCHEDA ARDUINO COMINCIA A LAMPEGGIARE.

```

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}

```

UNO SKETCH, COME TUTTI I PROGRAMMI SCRITTI NEI VARI LINUGAGGI, E' UN ISIEME DI ISTRUZIONI PER UN COMPUTER. ESAMINANDO LO SKETCH BLINK VEDIAMO CHE CI SONO DUE SEZIONI PRINCIPALI, **SETUP** E **LOOP**.

SETUP: E' ESEGUITO UNA SOLA VOLTA QUANDO SI LANCIAMO IL PROGRAMMA

LOOP: SI RIPETE ALL'INFINITO

SONO ENTRAMBI BLOCCHI DI CODICE CHIAMATI **FUNZIONI** E SONO PRESENTI IN OGNI SKETCH. SONO DELIMITATI DALLE PARENTESI GRAFFE { }.

[HTTP://ARDUINO.CC/EN/REFERENCE/HOMEPAGE](http://arduino.cc/en/reference/homepage)



NEL SITO DI ARDUINO TROVI LA GUIDA DI RIFERIMENTO E ALTRE RISORSE PER IMPARARE IL LINUGAGGIO DI PROGRAMMAZIONE.

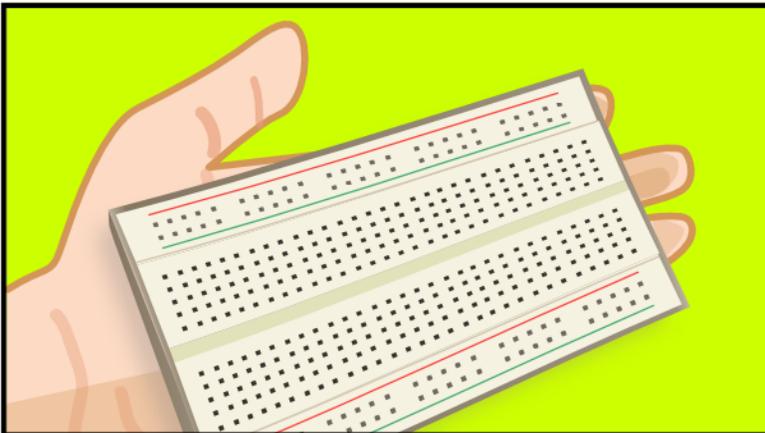
```

void setup() { //COMINCIA IL BLOCCO DI CODICE
  pinMode(13, OUTPUT); //PIN 13 COME OUTPUT
} //TERMINA IL BLOCCO DI CODICE

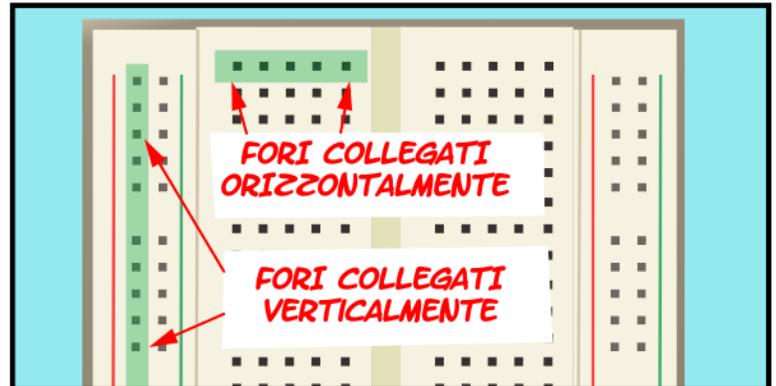
void loop() { //COMINCIA IL BLOCCO DI CODICE
  digitalWrite(13, HIGH); //IMPOSTA PIN 13 ALTO
  delay(1000); //ASPETTA 1 SECONDO
  digitalWrite(13, LOW); //IMPOSTA PIN 13 BASSO
  delay(1000); //ASPETTA 1 SECONDO
} //TERMINA IL BLOCCO DI CODICE

```

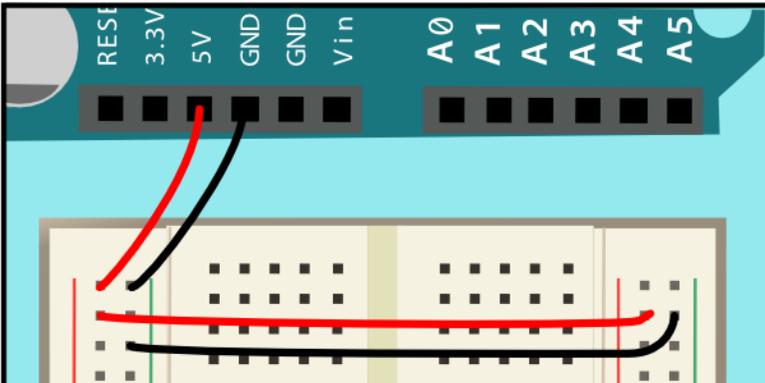
ESAMINIAMO IL PROGRAMMA LINEA PER LINEA E CERCHIAMO DI CAPIRE COSA FA OGNI ISTRUZIONE.



COME SI FA A CONTROLLARE OGGETTI CHE NON SONO SULLA SCHEDA ARDUINO? BISOGNA COLLEGARE LA SCHEDA A UNA **BREADBOARD** IN MODO DA POTER CREARE RAPIDAMENTE DEI CIRCUITI DI PROVA.



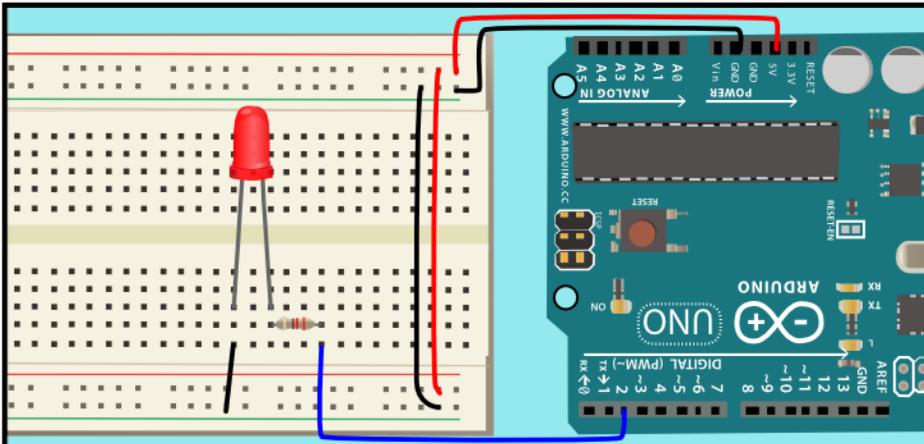
LA BREADBOARD HA DUE FILE DI FORI A SINISTRA E A DESTRA E CINQUE FILE DI FORI SU OGNUNO DEI LATI SEPARATI DALLA SCANALATURA CENTRALE. LE FILE ESTERNE SONO COLLEGATE VERTICALMENTE, QUELLE CENTRALI ORIZZONTALMENTE.



CON DEL FILO RIGIDO COLLEGHIAMO L'**ALIMENTAZIONE** E LA **MASSA** DELLA SCHEDA ARDUINO ALLE FILE COLLEGATE VERTICALMENTE. GLI ALTRI COMPONENTI VERRANNO COLLEGATI AI FORI CENTRALI E, QUANDO SERVE, ALL'ALIMENTAZIONE E ALLA MASSA.



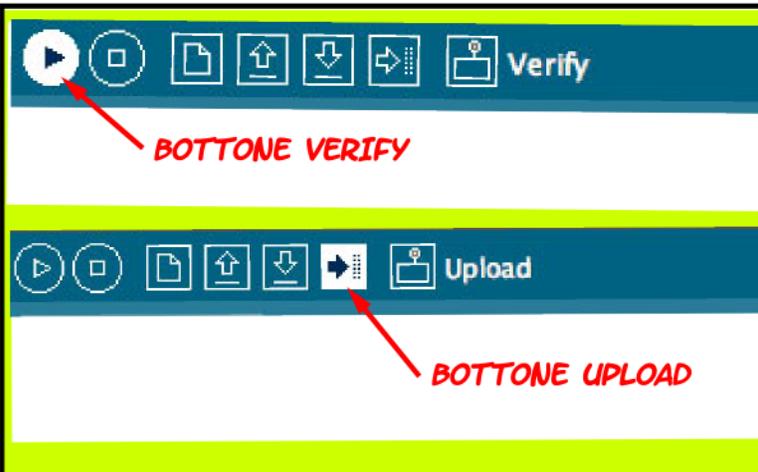
QUANDO LA CORRENTE CIRCOLA NELLA DIREZIONE GIUSTA UN **LED (DIODO AD EMISSIONE LUMINOSA)** SI ACCENDE. COLLEGHIAMO UN LED ALLA BREADBOARD E PROVIAMO A CONTROLLARLO SCRIVENDO DEL CODICE, CIOE' CON UN PROGRAMMA.



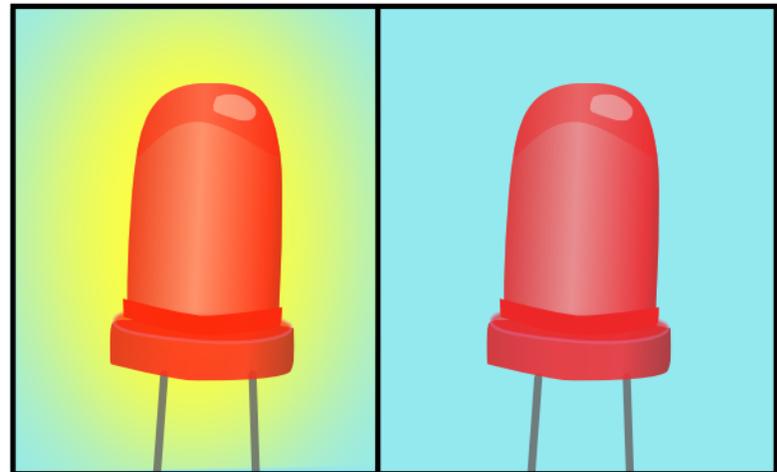
COLLEGHIAMO IL CATODO ALLA MASSA E, CON UNA RESISTENZA DA 220 OHM, L'ANODO AL PIN 2 DI ARDUINO. I PIN DA 2 A 13 POSSONO ESSERE IMPOSTATI COME INGRESSI O USCITE DIGITALI. CLICCA SUL BOTTONE **NEW** PER SCRIVERE UN NUOVO SKETCH.

```
void setup() {
  pinMode(2, OUTPUT);
}
void loop() {
  digitalWrite(2, HIGH);
  delay(500);
  digitalWrite(2, LOW);
  delay(500);
}
```

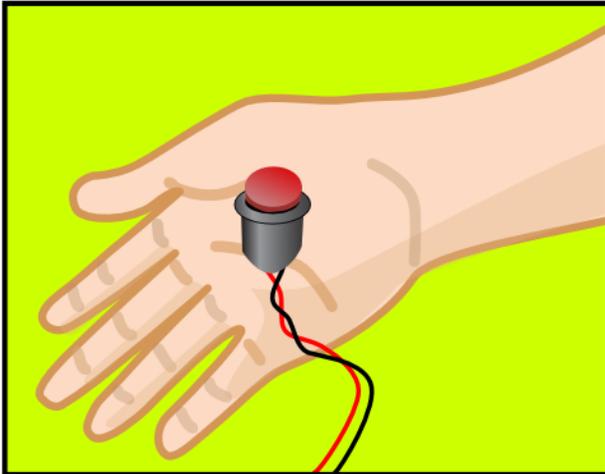
IN **SETUP** IMPOSTIAMO IL PIN 2 COME USCITA. IN **LOOP** ACCENDIAMO IL LED IMPOSTANDO IL PIN 2 ALTO E FACCIAMO UNA PAUSA DI 500 MILLISECONDI (MEZZO SECONDO) CON **DELAY**; POI SPEGNIAMO IL LED IMPOSTANDO IL PIN 2 BASSO E ASPETTIAMO UN ALTRO MEZZO SECONDO.



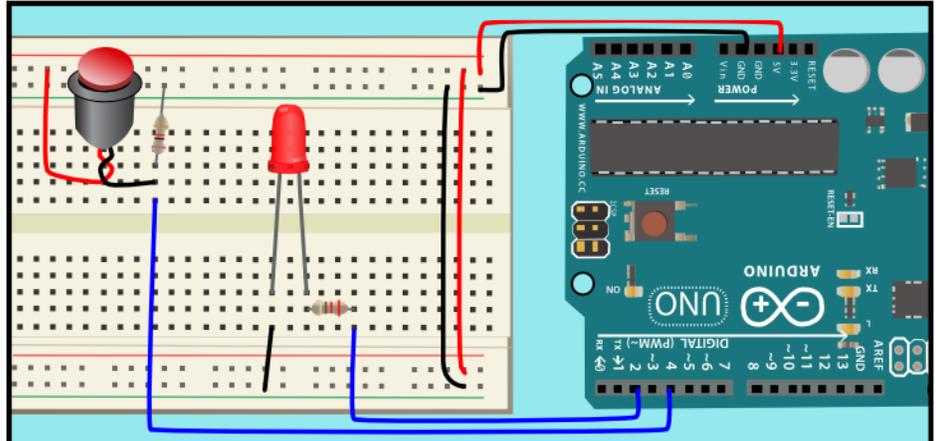
CLICCHIAMO **VERIFY** NEL MENU PER CONTROLLARE IL CODICE. SE NON CI SONO ERRORI CLICCHIAMO SU **UPLOAD** PER TRASFERIRE IL PROGRAMMA SU ARDUINO.



IL LED SI ILLUMINA PER MEZZO SECONDO, POI STA SPENTO PER UN ALTRO MEZZO SECONDO E RICOMINCIA DA CAPO.



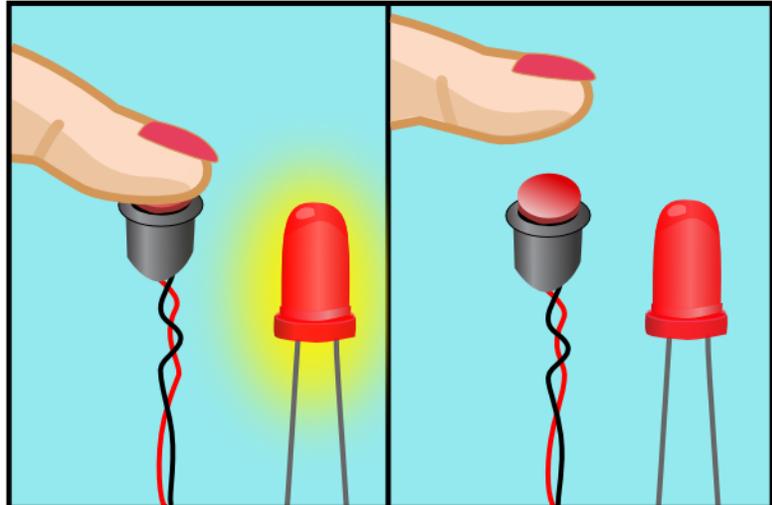
PROVIAMO AD AGGIUNGERE UN PULSANTE, CIOE' UN INGRESSO DIGITALE, PER ACCENDERE E SPEGNERE IL LED.



COLLEGHIAMO UN CAPO DEL PULSANTE AL PIN 4 DI ARDUINO E, CON UNA RESISTENZA DA 10K, ALLA MASSA. COLLEGHIAMO L'ALTRO CAPO ALL'ALIMENTAZIONE. IL LED E' SEMPRE CONNESSO AL PIN 2.

```
void setup() {  
  pinMode(2, OUTPUT);  
  pinMode(4, INPUT);  
}  
  
void loop() {  
  if(digitalRead(4)){  
    digitalWrite(2, HIGH);  
  }else{  
    digitalWrite(2, LOW);  
  }  
}
```

MODIFICHIAMO IL CODICE. IN SETUP DEFINIAMO IL PIN 2 COME USCITA E IL PIN 4 COME INGRESSO. IN LOOP USIAMO L'ISTRUZIONE IF (SE). SE IL PIN 4 E' ALTO IMPOSTIAMO IL PIN DEL LED ALTO, ALTRIMENTI (ELSE) LO IMPOSTIAMO BASSO, SPEGNENDO IL LED.

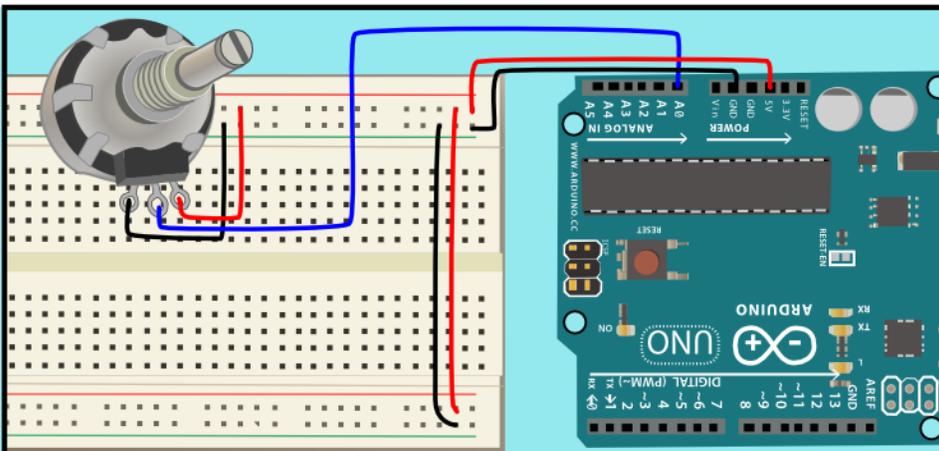


IL LED SI ACCENDE QUANDO PREMIAMO IL PULSANTE.

UN POTENZIOMETRO
E' UNA RESISTENZA VARIABILE.
IL VALORE DELLA RESISTENZA
CAMBIA RUOTANDO IL PERNO
NELLE DUE DIREZIONI.



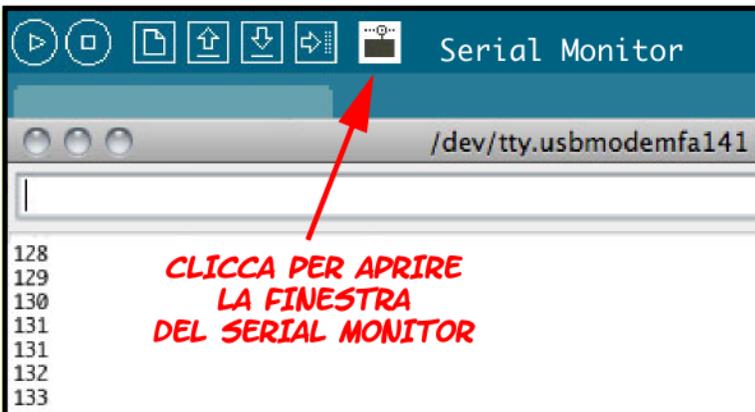
AGGIUNGIAMO UN INGRESSO ANALOGICO.
USEREMO UN POTENZIOMETRO.



COLLEGHIAMO IL PIN CENTRALE DEL POTENZIOMETRO AL PIN
ANALOGICO A0 DI ARDUINO. COLLEGHIAMO GLI ALTRI DUE PIN
ALL'ALIMENTAZIONE E ALLA MASSA.

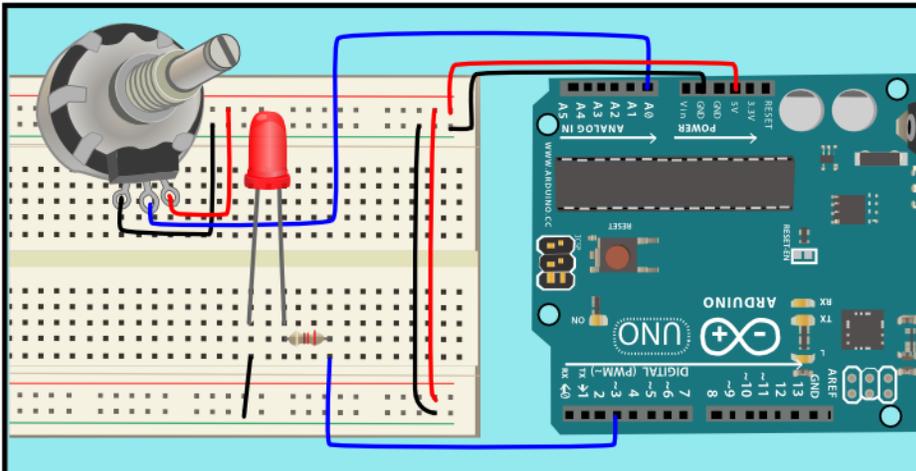
```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println(analogRead(A0));  
}
```

PER PRIMA COSA CONTROLLIAMO I VALORI CHE SI
OTTENGONO RUOTANDO IL POTENZIOMETRO CON IL
SERIAL MONITOR. IN SETUP IMPOSTIAMO LA
VELOCITA' DI TRASMISSIONE SERIALE (BAUD RATE)
A 9600. IN LOOP LEGGIAMO IL VALORE DAL PIN
ANALOGICO A0 E LO STAMPIAMO SUL SERIAL
MONITOR CON LA FUNZIONE **PRINTLN**.

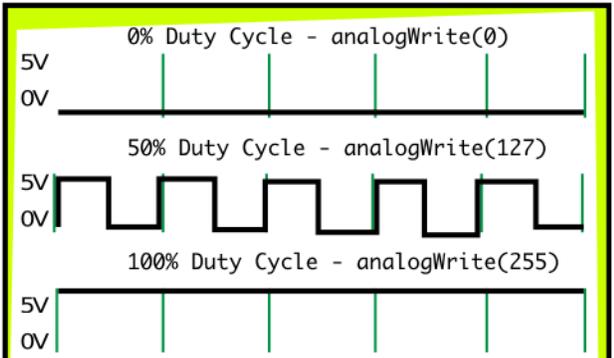


**CLICCA PER APRIRE
LA FINESTRA
DEL SERIAL MONITOR**

UNA VOLTA CARICATO IL PROGRAMMA SU ARDUINO
CLICCA SUL BOTTONE **SERIAL MONITOR** PER VEDERE
I VALORI CHE SI OTTEGGONO GIRANDO IL
POTENZIOMETRO. NELLA FINESTRA CHE SI APRE
VEDRAI CHE I VALORI VARIANO DA 0 A 1023.



USEREMO I VALORI DEL POTENZIOMETRO PER CONTROLLARE L'INTENSITA' LUMINOSA DEL LED. COLLEGHIAMO IL CATODO ALLA MASSA E, CON UN RESISTORE, L'ANODO AL PIN 3 DELLA SCHEDA.



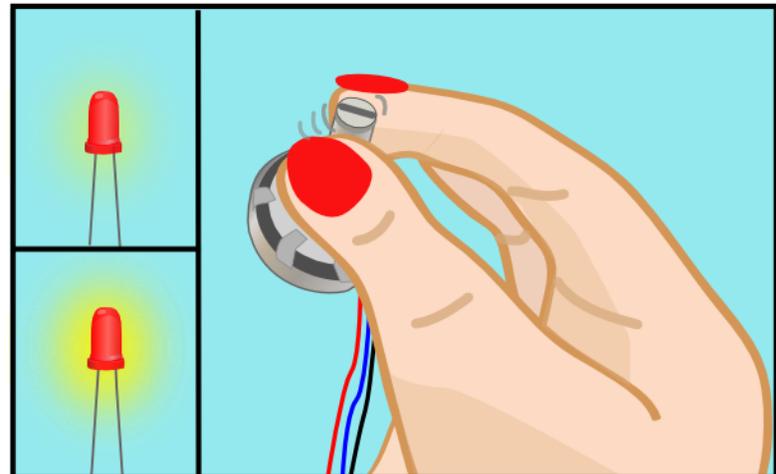
USIAMO LA MODULAZIONE A LARGHEZZA DI IMPULSO (PWM) PER SIMULARE UN VALORE ANALOGICO DI TENSIONE, PASSANDO DAL VALORE ALTO A QUELLO BASSO A INTERVALLI REGOLARI (DUTY CYCLE). SI PUO' USARE IL PWM CON I PIN 3, 5, 6, 9, 10 E 11.

```
int sensorValue = 0;

void setup() {
  pinMode(3,OUTPUT);
}

void loop() {
  sensorValue = analogRead(A0);
  analogWrite(3, sensorValue/4);
}
```

CREIAMO UNA VARIABILE PER MEMORIZZARE IL VALORE DEL POTENZIOMETRO. IN SETUP IMPOSTIAMO IL PIN 3 COME USCITA. IN LOOP LEGGIAMO E ASSEGNAMO IL VALORE DEL POTENZIOMETRO ALLA VARIABILE E LO USIAMO PER IMPOSTARE IL VALORE DEL PIN 3 COLLEGATO AL LED (DIVIDIAMO PER 4 PER AVERE UN INTERVALLO DI VALORI DA 0 A 255, CIOE' DI UN BYTE)



GIRANDO IL POTENZIOMETRO LA LUMINOSITA' DEL LED CAMBIA, DA COMPLETAMENTE SPENTO A MOLTO LUMINOSO.



LINK

SOFTWARE

SOFTWARE DOWNLOAD
[HTTP://WWW.ARDUINO.CC/EN/MAIN/SOFTWARE](http://www.arduino.cc/en/main/software)
LANGUAGE REFERENCE
[HTTP://ARDUINO.CC/EN/REFERENCE/HOMEPAGE](http://arduino.cc/en/reference/homepage)

FORNITORI

SPARKFUN ELECTRONICS
[HTTP://WWW.SPARKFUN.COM/](http://www.sparkfun.com/)
ADAFRUIT INDUSTRIES
[HTTP://ADAFRUIT.COM/](http://adafruit.com/)
MAKER SHED
[HTTP://WWW.MAKERSHED.COM/](http://www.makershed.com/)
JAMECO ELECTRONICS
[HTTP://WWW.JAMECO.COM/](http://www.jameco.com/)

GUIDE

ARDUINO SITE TUTORIALS
[HTTP://WWW.ARDUINO.CC/EN/TUTORIAL/HOMEPAGE](http://www.arduino.cc/en/tutorial/homepage)
LADY ADA
[HTTP://WWW.LADYADA.NET/LEARN/ARDUINO/INSTRUCTABLES](http://www.ladyada.net/learn/arduino/instructables)
[HTTP://WWW.INSTRUCTABLES.COM/TAG/TYPE-ID/CATEGORY-TECHNOLOGY/CHANNEL-ARDUINO/](http://www.instructables.com/tag/type-id/category-technology/channel-arduino/)

LIBRI

GETTING STARTED WITH ARDUINO BY MASSIMO BANZI
MAKING THINGS TALK: USING SENSORS, NETWORKS, AND ARDUINO TO SEE, HEAR, AND FEEL YOUR WORLD BY TOM IGOE
PHYSICAL COMPUTING: SENSING AND CONTROLLING THE PHYSICAL WORLD WITH COMPUTERS BY DAN O'SULLIVAN E TOM IGOE
ARDUINO COOKBOOK BY MICHAEL MARGOLIS

TESTO ORIGINALE E DISEGNI DI **JODY CULKIN**
PER ALTRE INFORMAZIONI VISITA [JODYCULKIN.COM](http://jodyculkin.com)

TRADOTTO IN ITALIANO DA LEONARDO CANDUCCI.

UN RINGRAZIAMENTO SPECIALE A TOM IGOE, MARIANNE PETIT, CALVIN REID, LA FACOLTA' E LO STAFF DELL'INTERACTIVE TELECOMMUNICATIONS PROGRAM ALLA NYU, IN PARTICOLARE DAN O'SULLIVAN, DANNY ROZIN E RED BURNS. GRAZIE A CINDY KARASEK, CHRIS STEIN, SARAH TEITLER, KATHY GONCHAROV E ZANNAH MARSH.

GRAZIE AL TEAM ARDUINO PER AVERCI DATO QUESTA PIATTAFORMA OPEN SOURCE ROBUSTA E FLESSIBILE.

E GRAZIE ALLA VIVACE COMUNITA' DI ARDUINO, SEMPRE ATTIVA E OGNI GIORNO PIU' GRANDE.

QUESTO DOCUMENTO - E L'ORIGINALE "INTRODUCTION TO ARDUINO" DI JODY CULKIN - SONO DISPONIBILI SECONDO LA LICENZA CREATIVE COMMONS ATTRIBUTION-NONCOMMERCIAL-SHAREALIKE 3.0 UNPORTED.

