

Corso di **Sistemi Interattivi**

Lezione 09. Arduino

Prof. Rudy Melli (rudymelli@ababrera.it)

www.vision-e.it/si

ACCADEMIA DI BELLE ARTI DI BRERA
Anno accademico 2018/2019

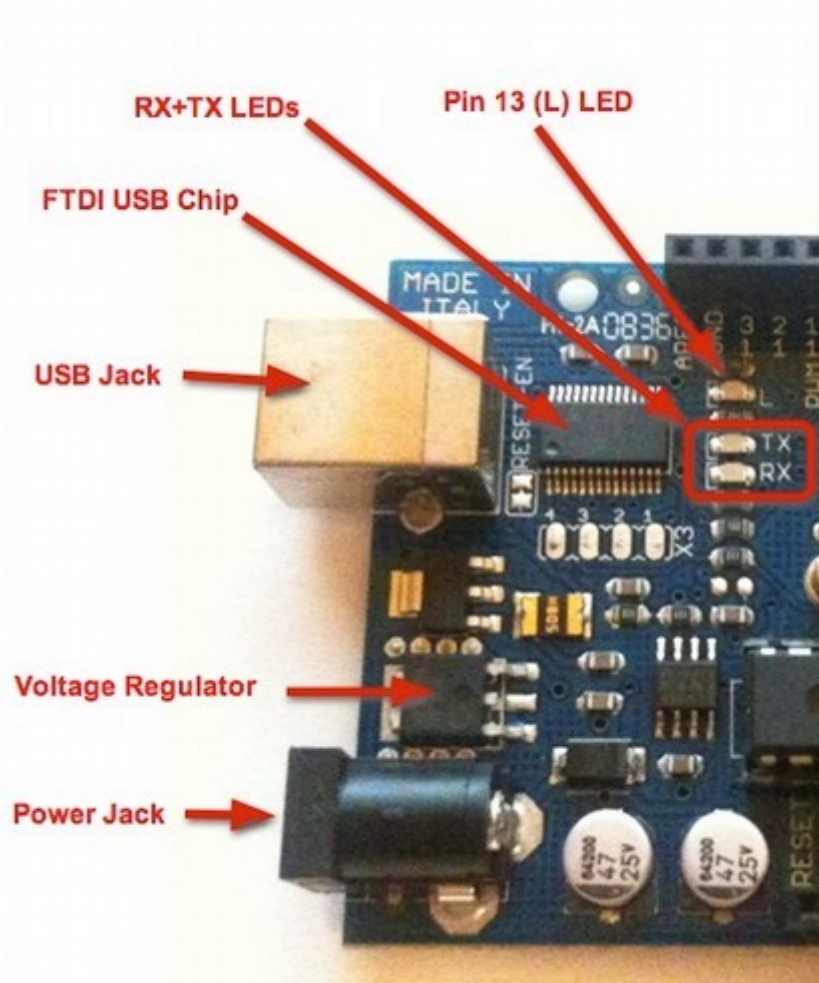
Arduino

Arduino è una scheda open source che permette la prototipazione rapida e l'apprendimento veloce dei principi fondamentali dell'elettronica e della programmazione.

Sviluppata presso l'Interaction Design Institute, un istituto di formazione post-dottorale con sede a Ivrea, fondato da Olivetti e Telecom Italia. Il nome della scheda deriva da quello di un bar di Ivrea (che richiama a sua volta il nome di Arduino d'Ivrea, Re d'Italia nel 1002) frequentato da alcuni dei fondatori del progetto.



Elementi Hardware



- Pin 13 Led, collegato al pin 13 di uscita
- RX-TX led, lampeggiano durante la comunicazione o durante l'esecuzione del programma
- USB Chip controller
- USB Jack – Segnale Pc e alimentazione 5V
- Voltage regulator
- Power Jack – Alimentatore 7-12V

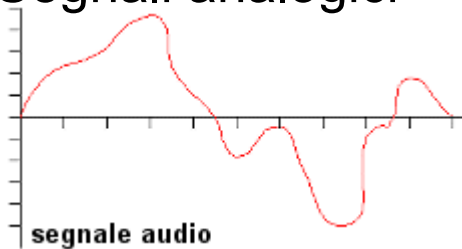
Teoria dei segnali per Arduino

Segnali digitali



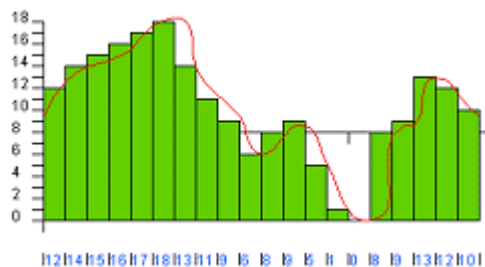
- Segnale binario, può assumere solo i valori 0 e 1
- Il valore 0 ha un voltaggio di 0 – 0.5 Volt
- Il valore 1 ha un voltaggio di 3.5 - 5 volt
- Un valore rappresenta un bit, 8 bit formano un byte.

Segnali analogici



- Rappresentano un valore che varia con continuità nel tempo
- Voltaggio compreso tra 0 e 5 Volt (solo valori positivi)

Conversione di segnali

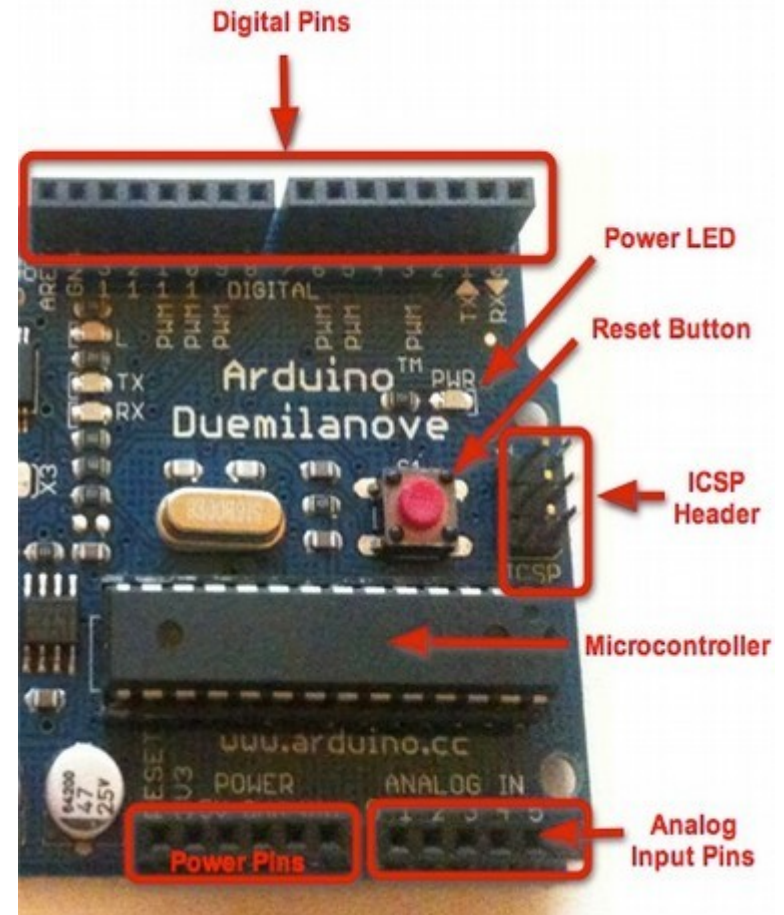


Da analogico a digitale: **Campionamento**: un segnale audio convertito in un segnale digitale

Da digitale ad analogico: **Modulazione o PWM (Pulse Width Modulation)**: la generazione di un'onda sinusoidale

Elementi Hardware

- Digital pin, sono 15 e possono essere (con alcune limitazioni)
 - Ingressi digitali
 - Uscite digitali
 - Uscite analogiche 0-5V
 - GND
- Analog Input pin
 - Ingressi analogici 0-5V
- Power Pin gnd-3.3-5-12 V
- Power Led
- Reset Button
- ICSP (porta seriale secondaria)
- Microcontroller ATmega328
 - Operating Voltage 5V
 - Analog Input Pins 6
 - DC Current per I/O Pin 40 mA
 - Flash Memory 32 KB
 - SRAM 2 KB / EEPROM 1 KB
 - Clock Speed 16 MHz



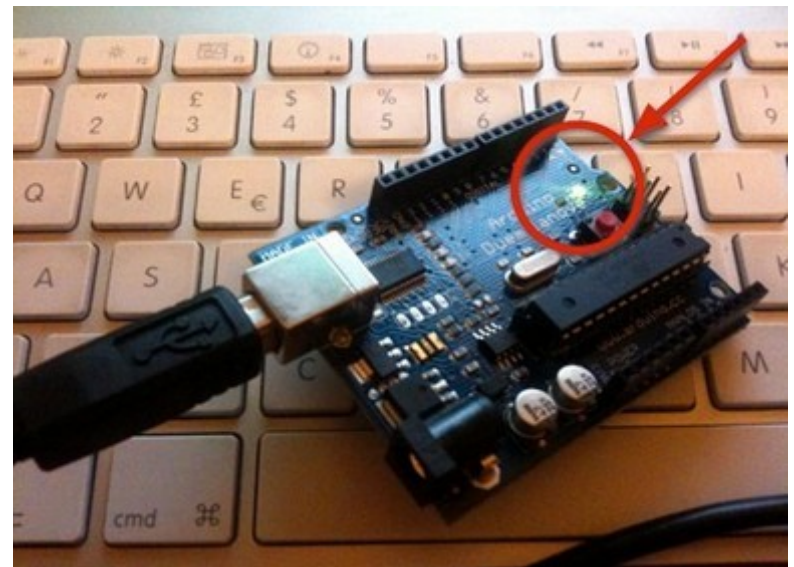
Riferimento: <http://www.maffucci.it/2010/01/30/lezione-01-incominciamo-con-arduino/>

Elettrotecnica





- ❏ Prima di continuare bisognerebbe apprendere le basi dell'elettrotecnica per poter rispondere a queste domande:
 - ☞ Cosa sono la tensione la corrente?
 - ☞ Cosa sono i Volt e gli Ampere?
 - ☞ Cosa sono una resistenza ed un condensatore?
 - ☞ Cos'è un circuito elettrico?
- ❏ Queste conoscenze non sono tutte fondamentali per poter usare Arduino (soprattutto usando circuiti o sensori da esempi online) ma vi aiuteranno a capire meglio il mondo di Arduino
- ❏ Ecco alcune risorse online:
 - ☞ <https://digilander.libero.it/nick47/icit01.htm>
 - ☞ <http://www.altratecnologia.com/arduino/>
 - <http://www.altratecnologia.com/wp-content/uploads/2014/11/Prima-lezione-arduino-base.pdf>
 - ☞ <https://www.progettiarduino.com/elettronica.html>

Ambiente di sviluppo

- ❏ Per poter programmare Arduino vi sono 2 modi:
 - ☞ Effettuare il download dell'ultima versione dell'ambiente di sviluppo e degli eventuali driver [consigliato]. Seguite le istruzioni del sito <https://www.arduino.cc/en/Main/Software>
 - ☞ Usare l'ambiente online Arduino Web Editor <https://create.arduino.cc/editor>
- ❏ A partire da Arduino Duemilanove la sorgente di alimentazione viene selezionata automaticamente e può avvenire tramite connettore USB o Jack di alimentazione. Appena alimentate la scheda si accenderà il led verde (PWR).



Ambiente di sviluppo

-  In Windows, la versione più semplice ed immediata è l'installer .exe che installa automaticamente anche i drivers necessari
-  Scaricando la versione zip, potrebbe essere necessario installare i driver presenti nella sotto-cartella drivers della cartella di Arduino come indicato nella guida ufficiale: <http://arduino.cc/en/Guide/Windows>
-  Selezionate la porta seriale da utilizzare per la comunicazione tra computer ed Arduino:
Tools > Serial Port menu.
Su Win selezionate la porta seriale connessa ad arduino (ad es. COM1)
Su Mac selezionate /dev/tty.usbserial-A9005fPz.
-  Selezionate il modello di Arduino che possedete
Tool > Board > Arduino Diecimila, Duemilanove, o Nano w/ATmega168
Ad Esempio Arduino Uno o Duemilanove con microcontrollore ATmega168

Arduino Web Editor

The screenshot displays the Arduino Web Editor interface in a web browser. The address bar shows the URL `https://create.arduino.cc/editor/rudymelli/sketch_may13a`. The left sidebar contains navigation options: EDITOR, Sketchbook, Examples, Libraries, Serial Monitor, Help, and Preferences. The 'Examples' section is expanded, showing a list of examples under categories like 01.BASICS (6), 02.DIGITAL (9), 03.ANALOG (6), and 04.COMMUNICATION (12). The 'Blink' example is selected. The main editor area shows the 'Blink' sketch with its code. The code is as follows:

```

1  /*
2   * Blink
3   * Turns on an LED on for one second, then off for one second, repeatedly.
4   *
5   * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
6   * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
7   * the correct LED pin independent of which board is used.
8   * If you want to know what pin the on-board LED is connected to on your Arduino model, check
9   * the Technical Specs of your board at https://www.arduino.cc/en/Main/Products
10  *
11  * This example code is in the public domain.
12  */
13  modified 8 May 2014
14  by Scott Fitzgerald
15
16  modified 2 Sep 2016
17  by Arturo Guadalupi
18
19  modified 8 Sep 2016
20  by Colby Newman
21  */

```

Below the code editor, a green success message states: "Success: Done uploading." Below this, a black terminal window shows the command used to compile and upload the sketch:

```

arduino-builder/arduino-builder -compile -core-api-version 10011 -build-path /tmp/889819699 -hardware arduino-
builder/hardware -hardware arduino-builder/packages/cores -tools arduino-builder/tools -tools arduino-
builder/packages/tools -built-in-libraries arduino-builder/latest -libraries /tmp/340048676/pinned -libraries
/tmp/340048676/custom -fqbn arduino:avr:uno -build-cache /tmp -verbose=false /tmp/340048676/Blink
Sketch uses 928 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048
bytes.
Flashing with command:C:/Users/melli/.arduino-create/arduino/avrdude/6.3.0-arduino9/bin/avrdude.exe -
CC:/Users/melli/.arduino-create/arduino/avrdude/6.3.0-arduino9/etc/avrdude.conf -q -q -patmega328p -carduino -
PCOM4 -b115200 -D -Uflash:w:C:/Users/melli/AppData/Local/Temp/arduino-create-agent280063972/Blink.hex:i

```

Arduino Web Editor

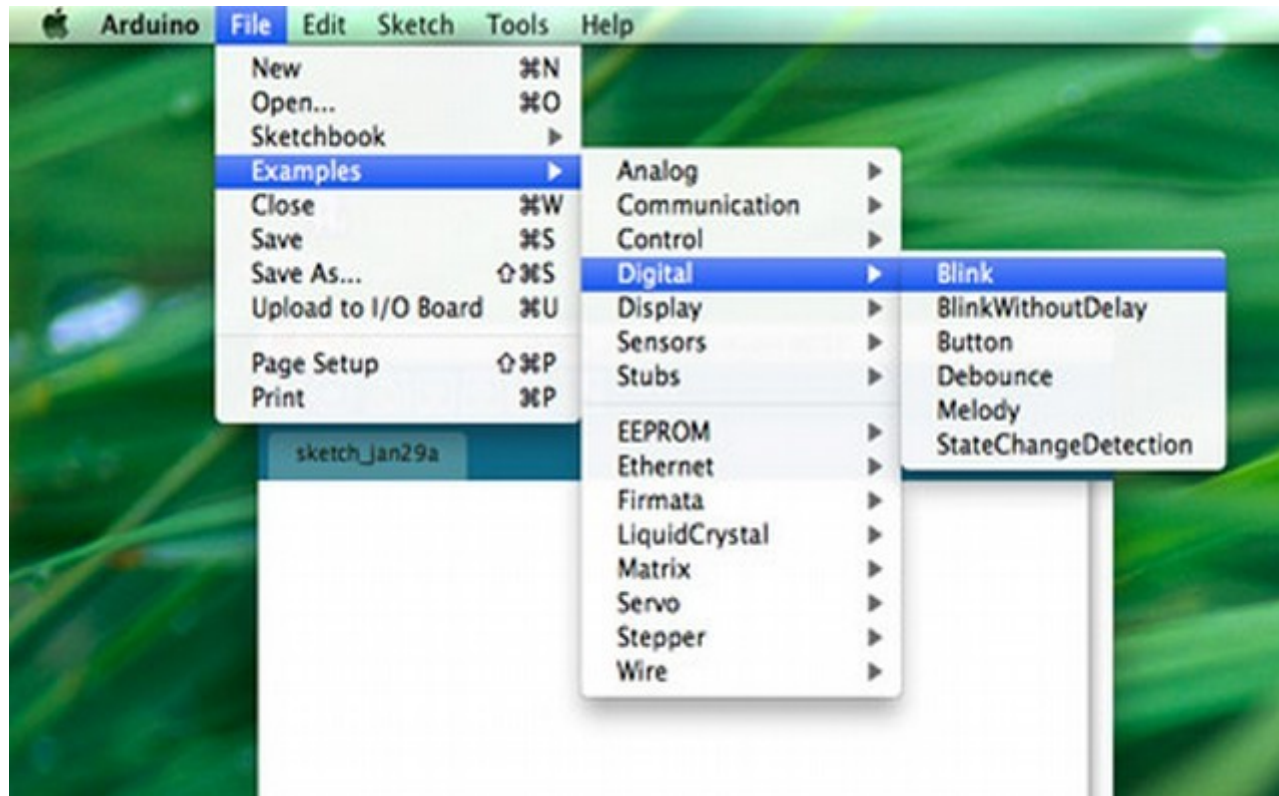
- ❏ In Windows è necessario installare un plugin che permetta all'editor web di vedere Arduino
 - ❏ Si utilizza con un semplice Web Browser connettendosi e registrandosi alla pagina <https://create.arduino.cc/editor>
 - ❏ Permette di avere un ambiente sempre aggiornato
 - ❏ E' alternativo all'ambiente classico installato direttamente sul PC
 - ❏ Funzionano gli stessi sketch
 - ❏ **E' necessaria una connessione internet per utilizzarlo**
-
- ❏ Nel corso faremo riferimento alla versione classica da installare perché non necessita di una connessione internet per lavorare

Ambiente di sviluppo

Eseguire l'upload di un programma

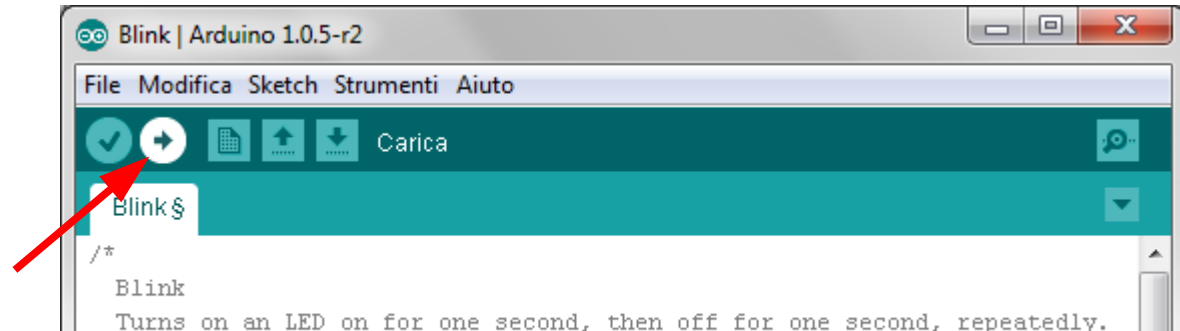
Aprire l'esempio "Blink" che si trova in:

File > Examples > Digital > Blink



Ambiente di Sviluppo

- ❏ Ora fate l'upload del file "Blink" su Arduino facendo click sul pulsante "Carica/Upload"



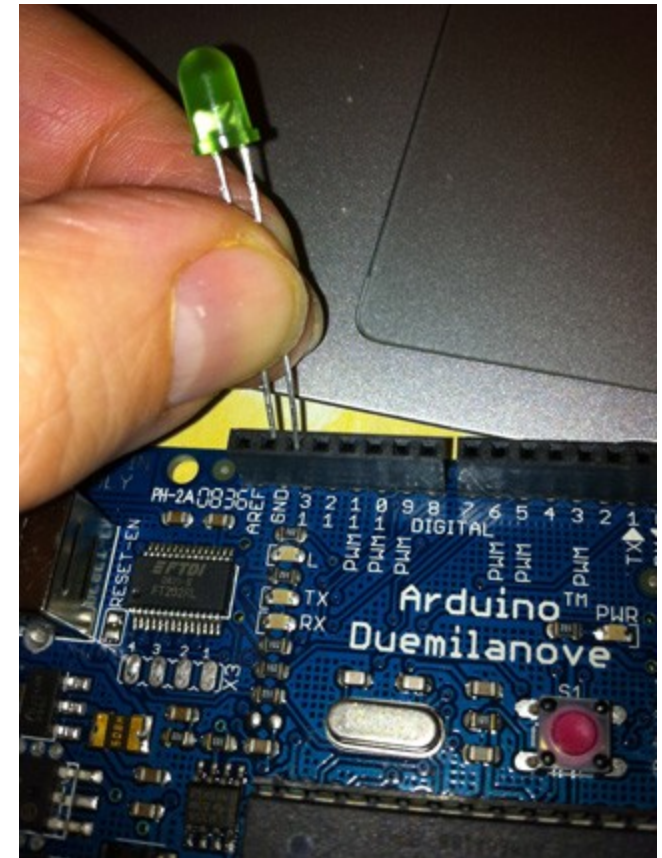
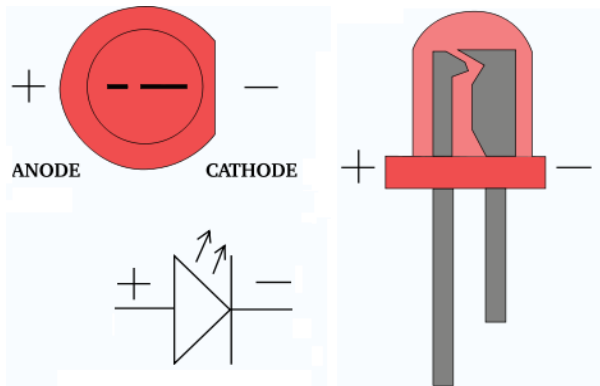
ci vorrà qualche secondo, durante questa operazione i led RX e TX (ricezione e trasmissione) lampeggiano, se tutto andrà a buon fine vi verrà restituito il messaggio "Done uploading." nella status bar.

- ❏ Appena terminato l'upload su Arduino il pin 13 (L) corrispondente al led arancione, incomincia a lampeggiare.



Il Primo Programma: Blink

- ❏ Questo programma di esempio servirà a far lampeggiare un led.
- ❏ Collegare un led tra il pin 13 e GND. Rispettare la polarità del led, il catodo collegato a (-)(GND) e l'anodo al (+)(13)



Blink: lo sketch



Aprire il programma blink

File > Examples > 1.Basics > Blink



Oppure create all'interno dell'editor il seguente programma:

```
// Esempio 01: far lampeggiare un led
```

```
#define LED 13          // LED collegato al pin digitale 13
```

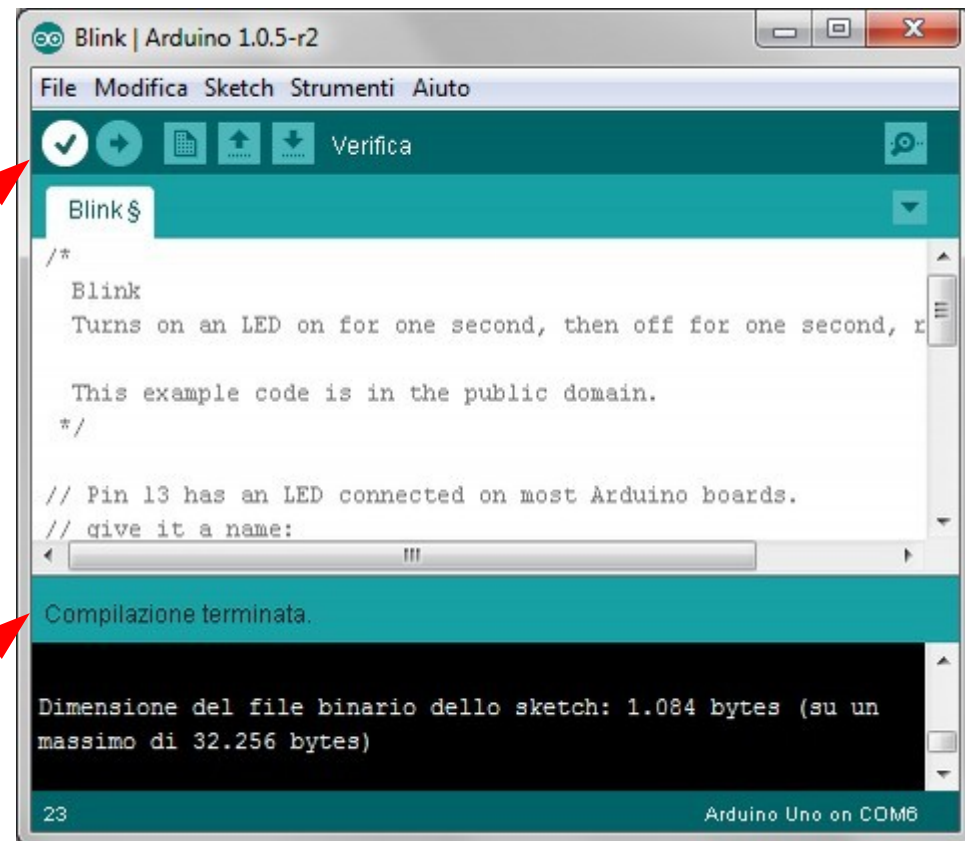
```
void setup() {  
  pinMode(LED, OUTPUT); // imposta il pin digitale come output  
}
```

```
void loop() {  
  digitalWrite(LED, HIGH); // accende il LED  
  delay(1000);             // aspetta un secondo  
  digitalWrite(LED, LOW);  // spegne il LED  
  delay(1000);             // aspetta un secondo  
}
```


Blink: Verifica

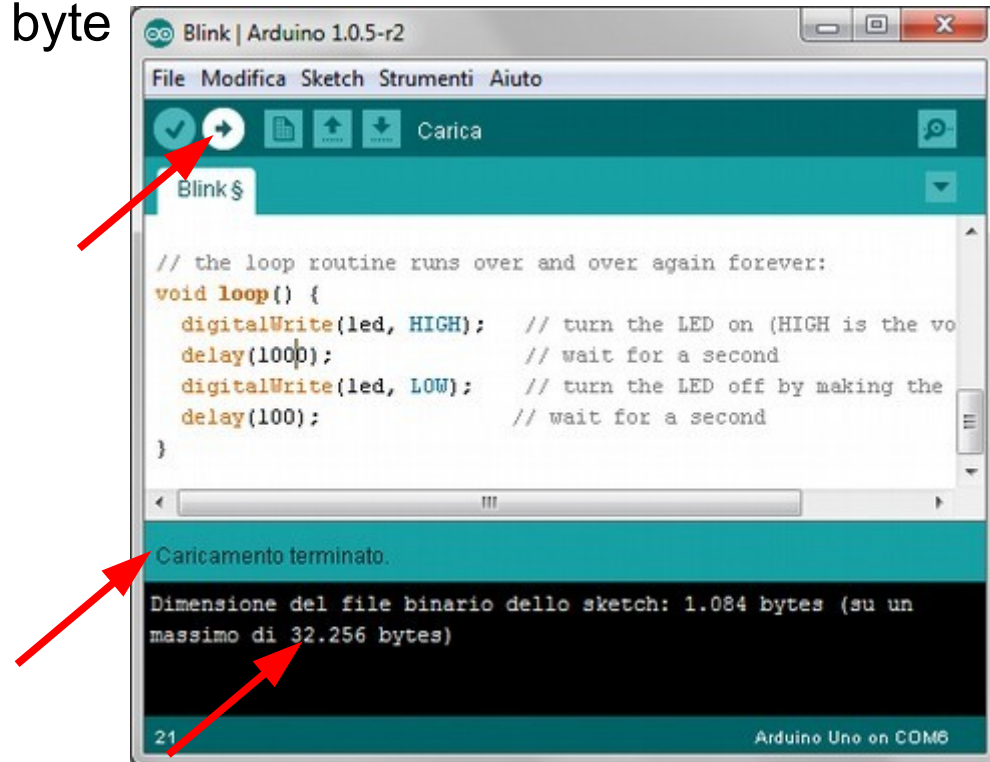
❏ Ora occorre verificare il programma facendo click su “Verify” e se tutto è corretto in basso all’IDE compare il messaggio “Done compiling” ciò indica che l’IDE Arduino ha tradotto lo sketch in un programma eseguibile sulla scheda Arduino.

❏ In caso contrario apparirà la lista degli errori da risolvere



Blink: Upload e Run

- ❏ Ora occorre caricare sulla scheda il vostro programma: facendo click su “Carica/Upload” la scheda interrompe l’esecuzione di ciò che stava eseguendo e carica il nuovo programma in memoria
- ❏ Sulla scheda i led identificati con RX e TX lampeggiano, così come ogni volta che viene inviato un byte
- ❏ Il nuovo programma viene immediatamente messo in esecuzione.
- ❏ In questo caso il led comincia a lampeggiare.



Programmare Arduino

- Definizione delle costanti: generalmente vengono definite le costanti riguardanti pin di I/O e parametri del sistema (es periodo del lampeggio)

```
#define LED 13          // LED collegato al pin digitale 13
```

- Funzione setup(). Viene richiamata ad inizio programma ovvero quando viene acceso il dispositivo. Contiene le istruzioni di inizializzazione che devono essere eseguite una sola volta.

```
void setup() {  
    pinMode(LED, OUTPUT);    // imposta il pin digitale come output  
}
```

- Funzione loop(). E' la funzione di lavoro. Sarà richiamata in modo ciclico all'infinito. Contiene le istruzioni che gestiscono la dinamica del sistema.

```
void loop() {  
    digitalWrite(LED, HIGH); // accende il LED  
    delay(1000);             // aspetta un secondo  
    digitalWrite(LED, LOW);  // spegne il LED  
    delay(1000);             // aspetta un secondo  
}
```

Seriale

- ❏ Quando con Upload si carica un programma su Arduino, questo diventerà subito attivo e sarà eseguito sulla scheda, non sul PC
- ❏ La comunicazione con il PC da parte di Arduino si può fare attraverso la connessione USB (il cavo quindi deve rimanere collegato al PC), oppure se necessaria, con schede di espansione Ethernet o wifi. In ogni caso la comunicazione avviene scrivendo e leggendo sulla porta Seriale
- ❏ Un modo per verificare il funzionamento sul PC è inserire nel codice Arduino, delle scritture su porta seriale, ed utilizzare il
Monitor Seriale
presente sulla GUI, per visualizzare il testo in uscita dalla scheda
- ❏ Arduino comunicherà con l'esterno (quindi anche con Processing) scrivendo e leggendo dalla porta seriale con `Serial.write` e `Serial.read`
- ❏ In Processing esistono librerie di sincronizzazione che semplificano la vita in caso di utilizzo di molte variabili: `VSync` e `AP-Sync` ognuna con la sua sintassi specializzata

Leggere/Scrivere i dati da Processing

 Si usa la comunicazione seriale

Arduino	Processing
<pre>void setup() { //inizializza comunicaz. Seriale con un baudrate di 9600 Serial.begin(9600); } void loop() { //Invio una stringa sulla porta seriale Serial.println("17"); //aspetto 100 ms per impedire troppi invii delay(100); }</pre>	<pre>Serial myPort; // Variabile che gestisce la seriale String val; // Dati ricevuti dalla porta seriale void setup() { // Inserire la porta che sta usando arduino String portName = Serial.list()[0]; // Impostare il medesimo baudrate di Arduino myPort = new Serial(this, portName, 9600); } void draw() { if (myPort.available() > 0) { // Se ci sono dei dati disponibili // li leggo e memorizzo in 'val' val = myPort.readStringUntil('\n'); } println(val); //printo il valore nella console }</pre>

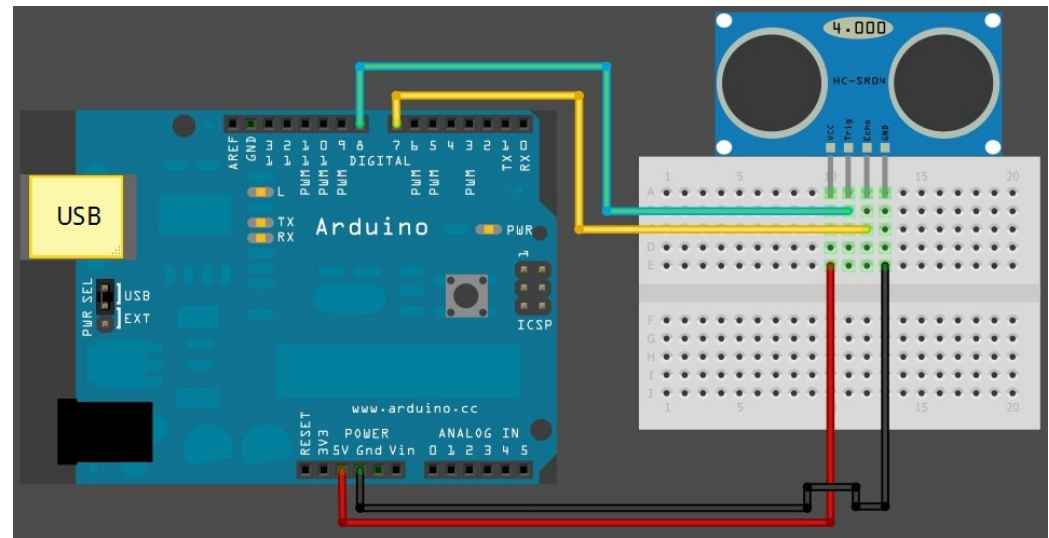
Cosa viene scritto in Processing?

 Fare riferimento a questo guida per approfondimenti

 <https://learn.sparkfun.com/tutorials/connecting-arduino-to-processing>

Distanza

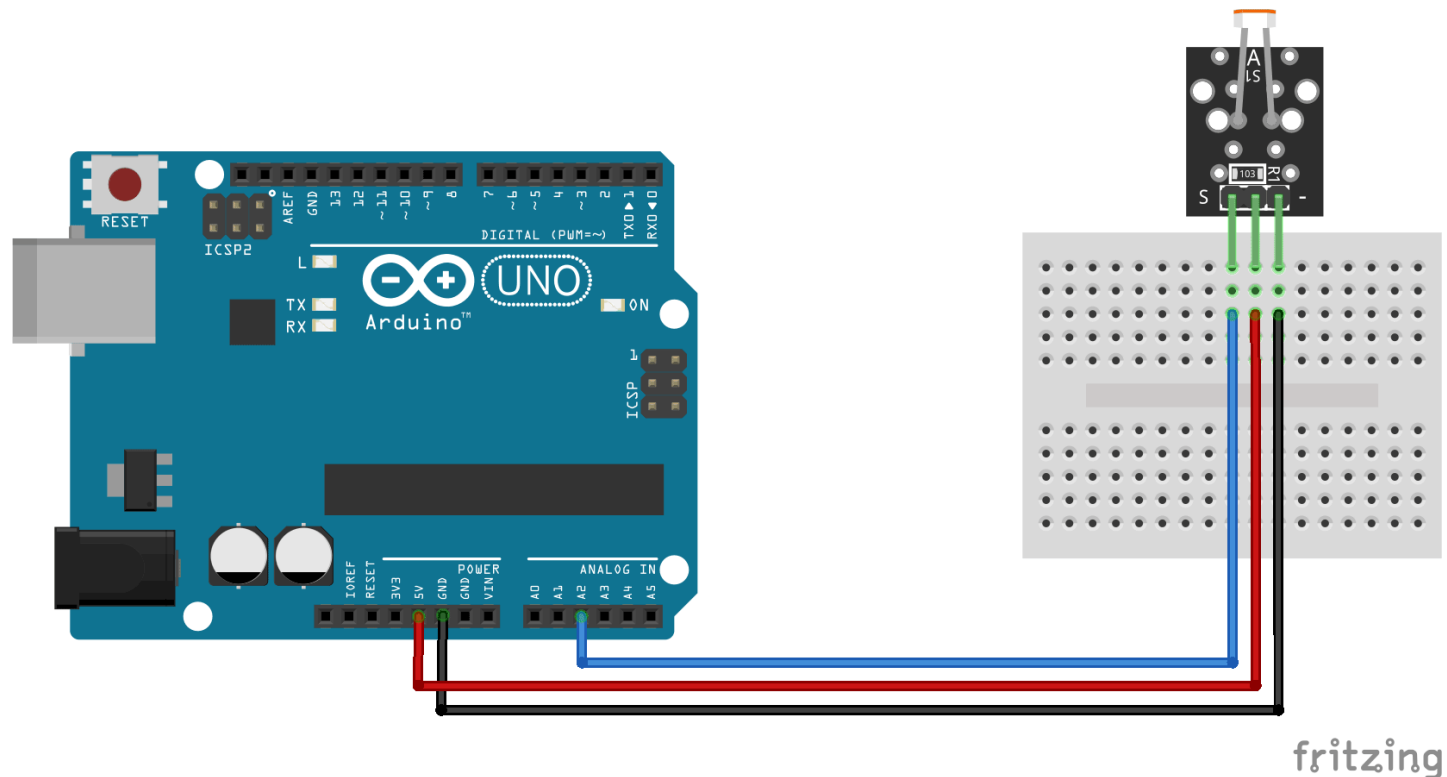
- Uno dei sensori distanza più famosi ed economici per Arduino è HC-SR04, a tecnologia ultrasonica (legge quanto ci mette un segnale audio a tornare indietro). Visibilità radiale di circa 90 gradi



- Esempio completo su Gdrive nella cartella Sketches\Arduino\HCSR04
- Tramite la comunicazione seriale possiamo visualizzare in Processing cosa “vede” il sensore

Leggere una grandezza analogica: luminosità

- Progetto di un sistema per leggere il valore di un sensore analogico, in particolare da un sensore di luminosità



Luminosità: lo sketch



Aprire il programma blink

File > Examples > 3.Analog > AnalogInput



Oppure create all'interno dell'editor il seguente programma:

```
int sensorPin = A0;           // select the input pin for the potentiometer
int ledPin = 13;              // select the pin for the LED
int sensorValue = 0;          // variable to store the value coming from the sensor








void setup() {
  pinMode(ledPin, OUTPUT);    // declare the ledPin as an OUTPUT:
}

void loop() {
  sensorValue = analogRead(sensorPin);    // read the value from the sensor: [0-1024]

  digitalWrite(ledPin, HIGH);             // turn the ledPin on
  delay(sensorValue);                     // stop the program for <sensorValue> millisec:

  digitalWrite(ledPin, LOW);              // turn the ledPin off:
  delay(sensorValue);                     // stop the program for for <sensorValue> millisec
}
```

Risorse online

-  <https://www.arduino.cc>
-  Guide direttamente dal forum Arduino Forum > International > Italiano > Megatopic > LINK UTILI:
<http://forum.arduino.cc/index.php?topic=126861.0>
- 
-  <http://www.progettiarduino.com/progetti-e-tutorial.html>
-  <https://learn.sparkfun.com/tutorials/connecting-arduino-to-processing>
-  Sensori (da verificare): <https://www.homotix.it/vendita/sensori>
-  Kuman, Super Starter Kit → kit per iniziare a basso costo (ebay)

Esercizi

- Far lampeggiare il led al pin 13 della scheda
- Collegare Arduino con Processing. Utilizzare un sensore esterno, come temperatura, pressione, presenza, ecc... e visualizzare il suo valore in tempo reale nella finestra di Processing
- Cambiare il colore dello sfondo della finestra di Processing in funzione del valore del sensore precedentemente collegato
- Comandare una lampadina collegata alla 220V con Arduino grazie all'uso di un relè, cambiando lo stato (acceso/spento) da Processing.
 - ☞ Vedi questo esempio come spunto:
<http://www.circuitbasics.com/setting-up-a-5v-relay-on-the-arduino/>