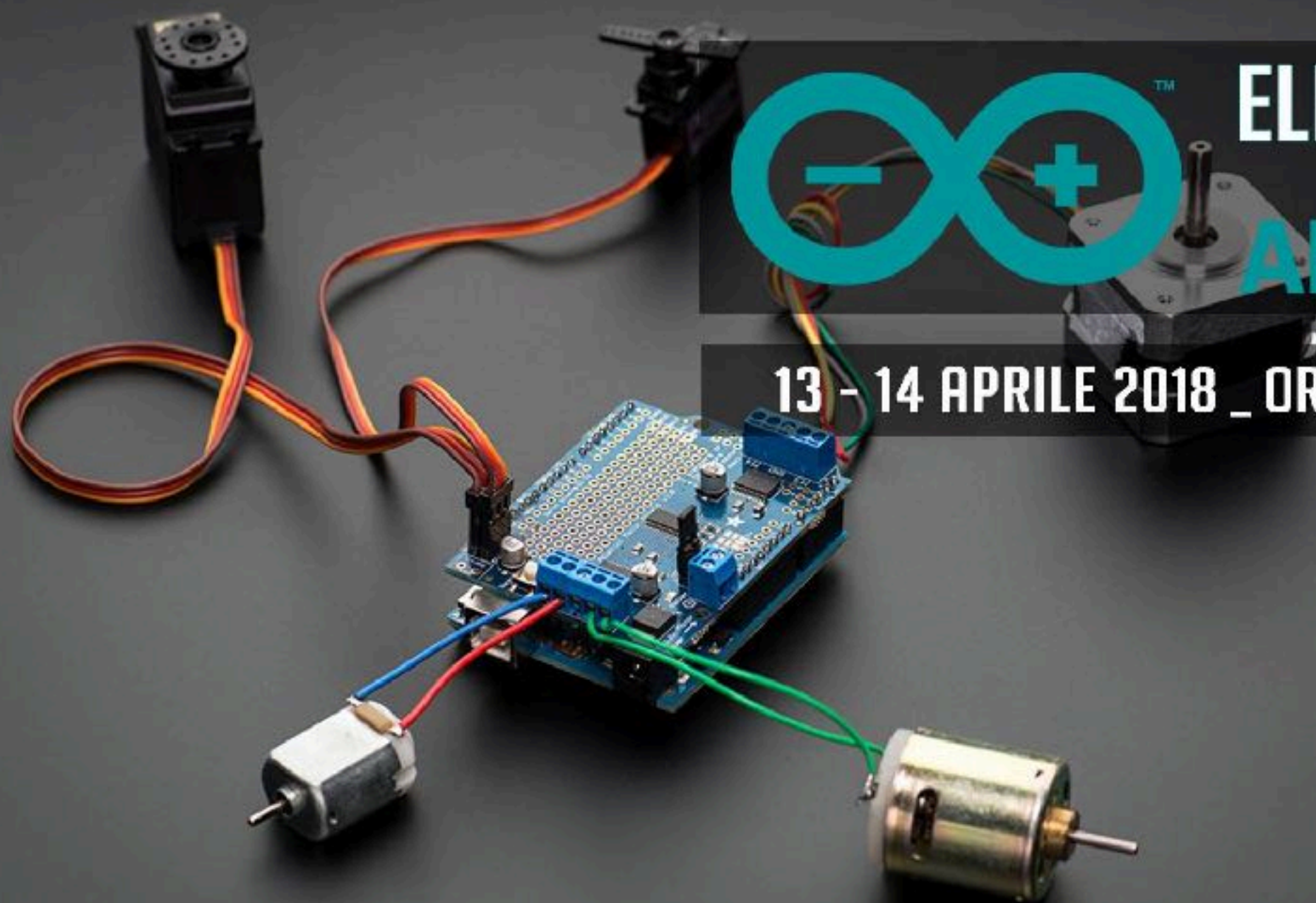


ELETTRONICA CON ARDUINO

13/14 aprile 2018



ELETTRONICA
CON
ARDUINO

13 - 14 APRILE 2018 _ ORE: 16.00 - 19.30

Giorno 2

- Arduino IDE: introduzione, installazione e interfaccia;
- Cosa sono gli sketch: commenti, funzioni, variabili, istruzioni, strutture di controllo, librerie.
- Componenti del kit;
- Realizzazione primo esempio: blink;
- Esercitazioni.

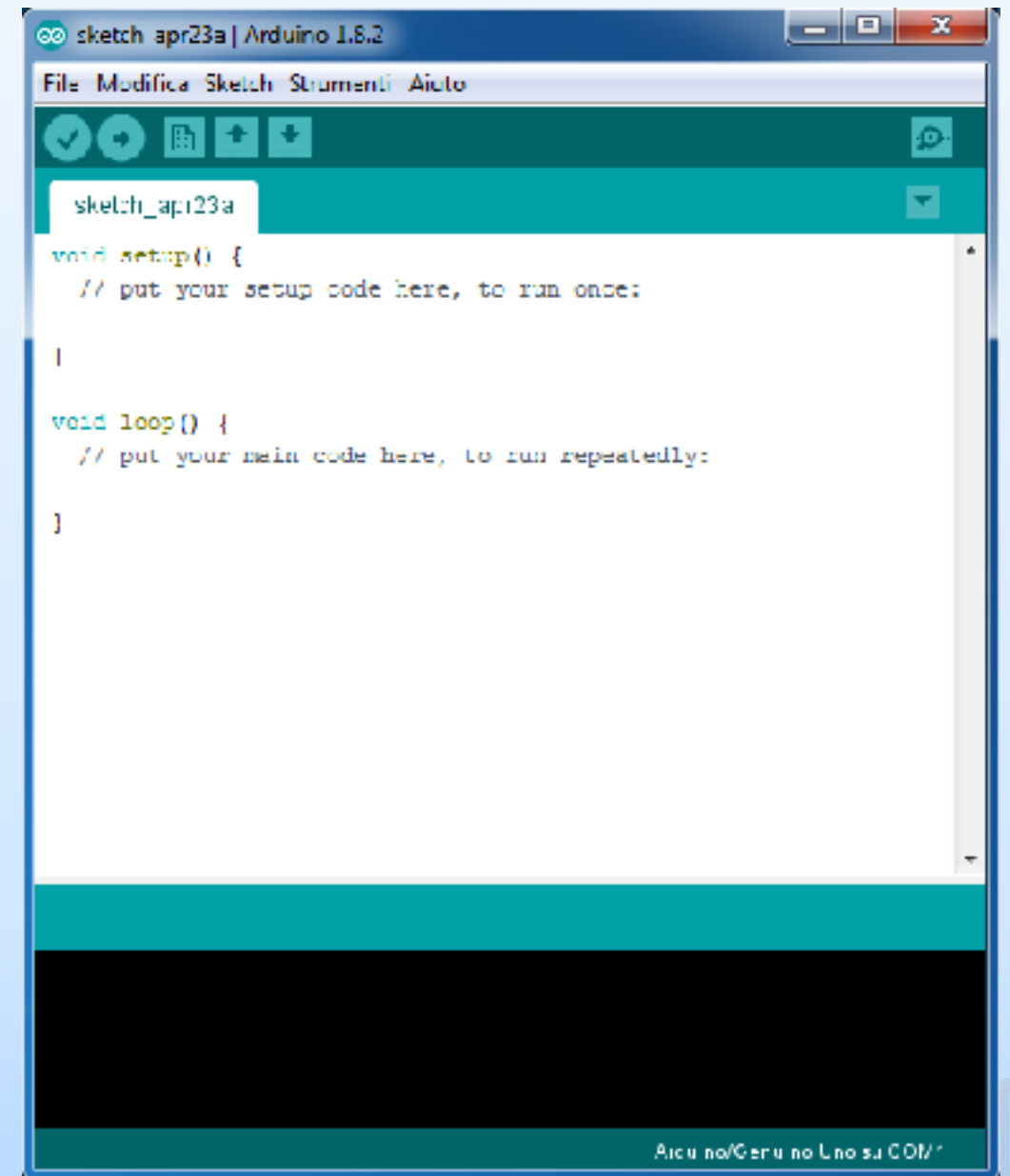


ARDUINO IDE

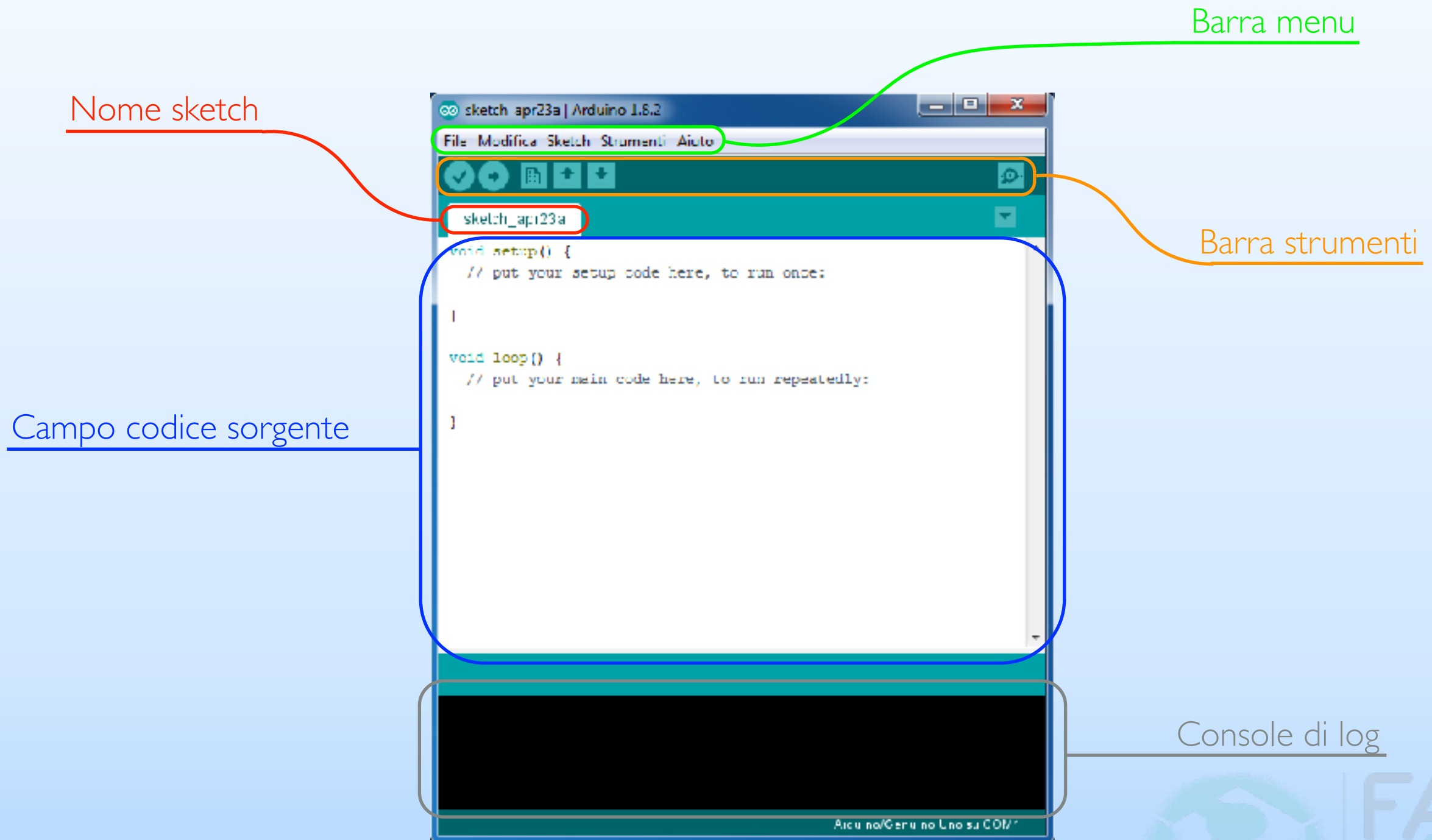
[DOWNLOAD](#)

- L'Arduino IDE (**I**ntegrated **D**evelopment **E**nvironment) è un'applicazione multiplatforma che consente la stesura del codice sorgente (sketch).
- Include un editor di testo dotato di **syntax highlighting**, controllo delle parentesi e indentazione automatica.
- Con un singolo click è in grado di **compilare** e caricare il programma su qualsiasi scheda Arduino.

processo durante il quale uno sketch viene convertito in linguaggio macchina, comprensibile per il microcontrollore.



Arduino IDE INTERFACCIA

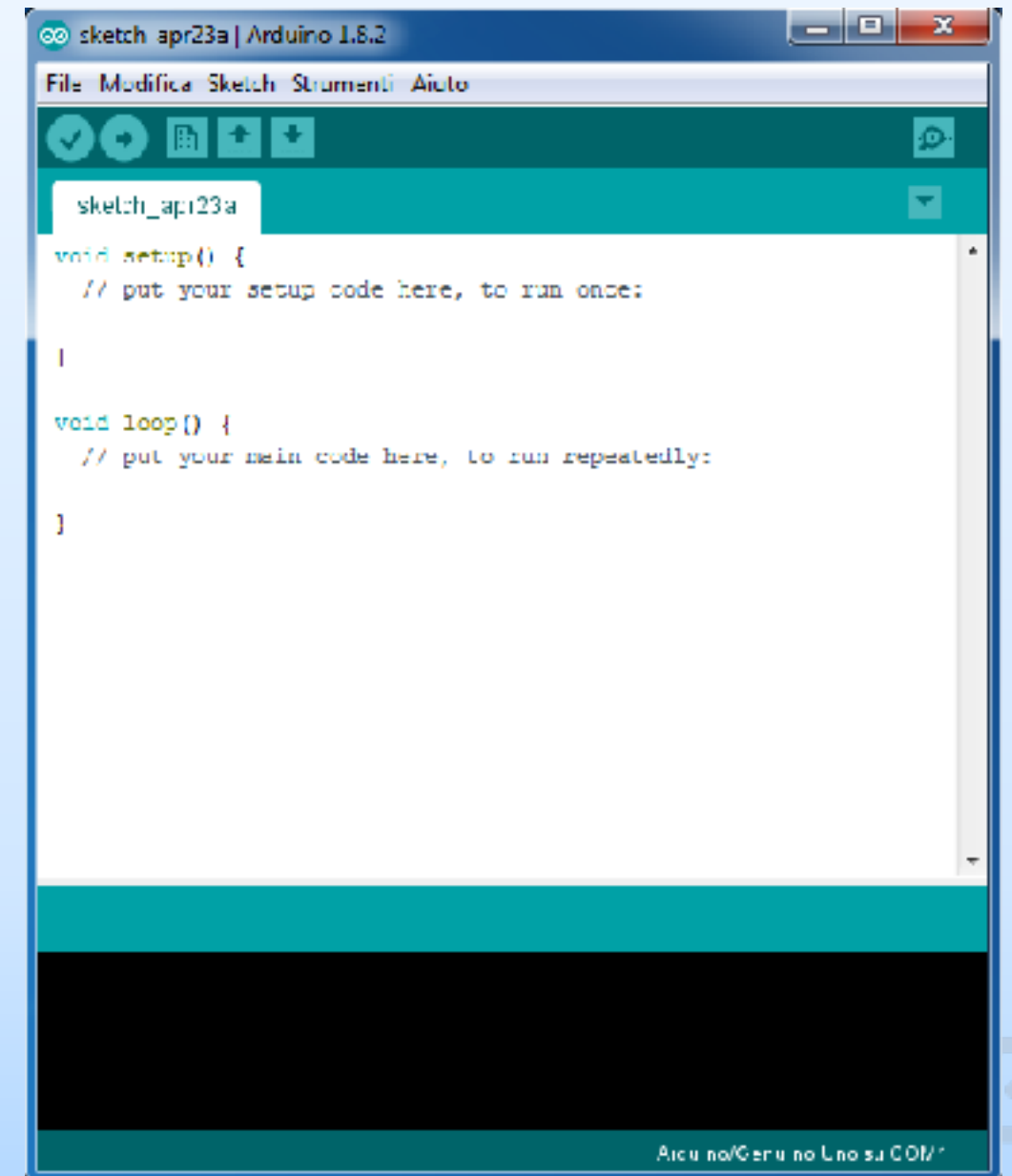


COSA SONO GLI SKETCH

Sketch è il nome che Arduino dà ai suoi programmi; i file contenenti gli sketch hanno estensione **.ino**.

Funzione: blocco di codice, racchiuso da una coppia di parentesi graffe, richiamabile attraverso uno specifico nome.

- Lo sketch base contiene le due funzioni principali:
- **setup()**: le istruzioni contenute in questa funzione vengono eseguite una sola volta all'accensione di Arduino.
- **loop()**: le istruzioni contenute in questa funzione vengono eseguite ciclicamente fino allo spegnimento di Arduino.

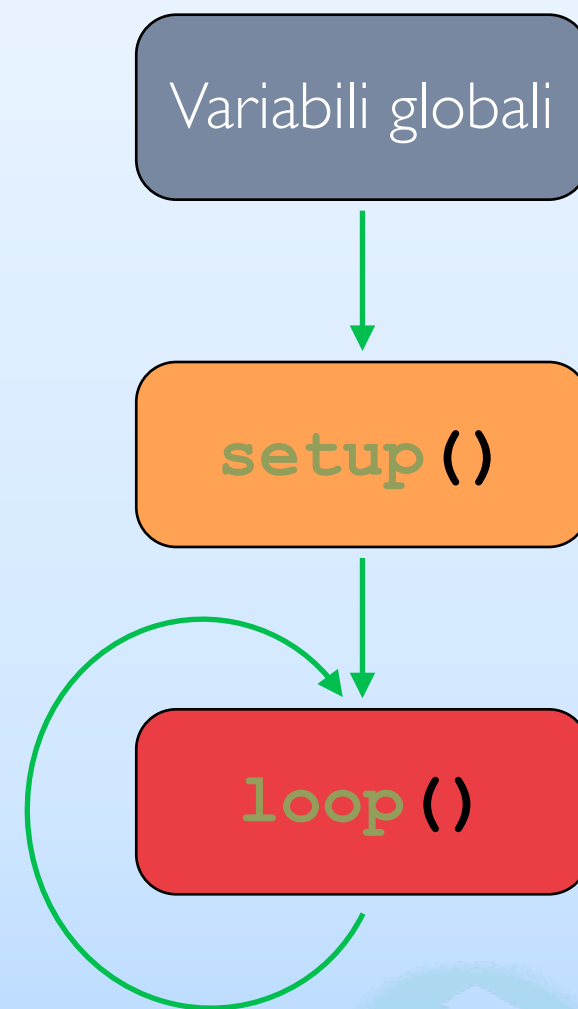


COSA SONO GLI SKETCH

Sketch è il nome che Arduino dà ai suoi programmi; i file contenenti gli sketch hanno estensione **.ino**.

Funzione: blocco di codice, racchiuso da una coppia di parentesi graffe, richiamabile attraverso uno specifico nome.

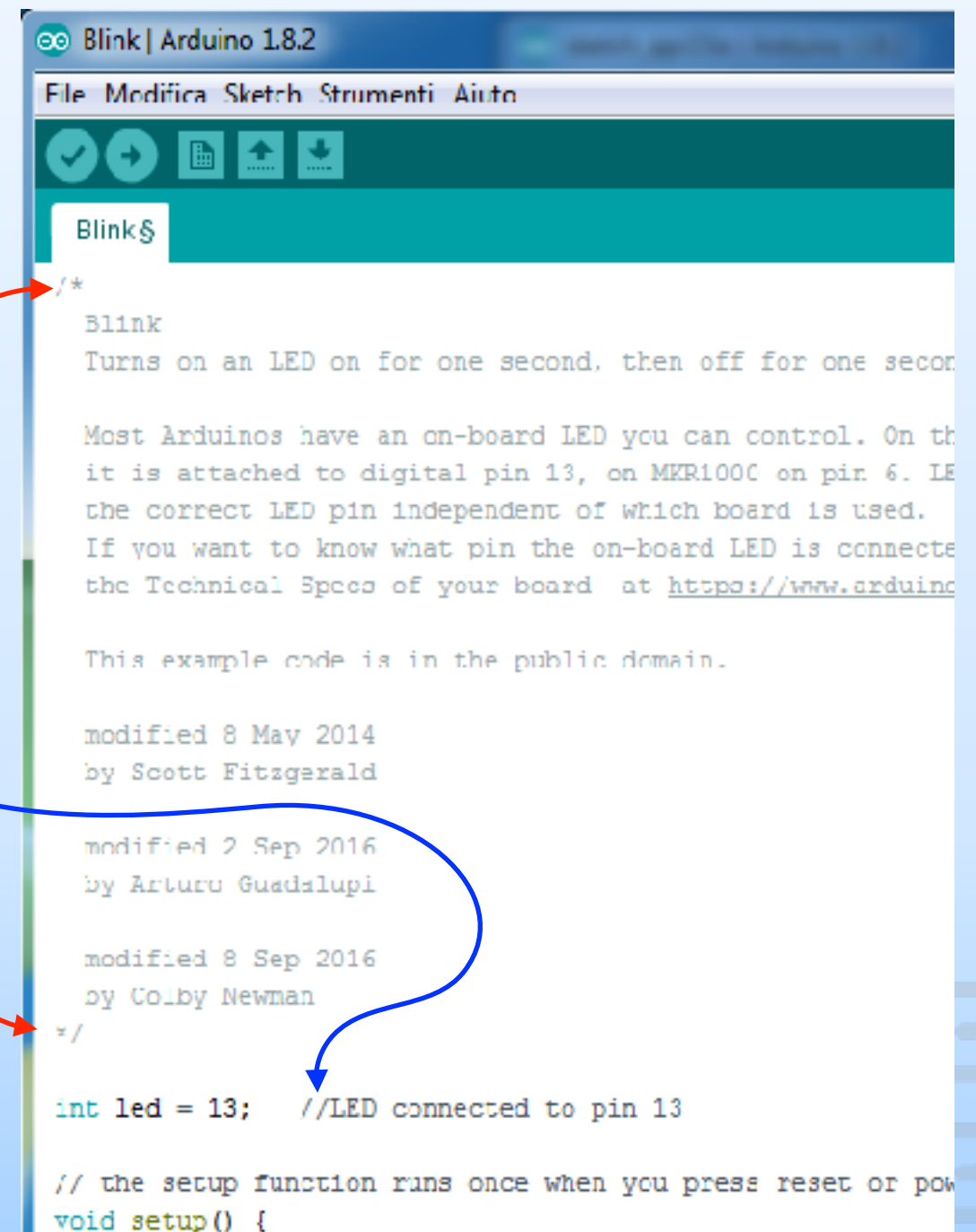
- Lo sketch base contiene le due funzioni principali:
- **setup()**: le istruzioni contenute in questa funzione vengono eseguite una sola volta all'accensione di Arduino.
- **loop()**: le istruzioni contenute in questa funzione vengono eseguite ciclicamente fino allo spegnimento di Arduino.



COSA SONO GLI SKETCH

Commenti

- Un commento è una parte di testo che non verrà compilata;
- Su più linee: `/* questo è un commento */`
- Su una linea: `//questo è un commento`



The screenshot shows the Arduino IDE interface with the 'Blink' sketch open. The menu bar includes 'File', 'Modifica', 'Sketch', 'Strumenti', and 'Aiuto'. The toolbar contains icons for checking, running, saving, and uploading. The sketch content is as follows:

```
/*  
  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  
  Uno board it is attached to digital pin 13, on the MKR1000 on pin 6. LED  
  pins are usually marked with LEDs on the PCB. To avoid confusion  
  the correct LED pin independent of which board is used.  
  If you want to know what pin the on-board LED is connected to  
  check the Technical Specs of your board at https://www.arduino.cc/en/Tutorial/Blink  
  
  This example code is in the public domain.  
  
  modified 8 May 2014  
  by Scott Fitzgerald  
  
  modified 2 Sep 2016  
  by Arturo Guadalupi  
  
  modified 8 Sep 2016  
  by Colby Newman  
*/  
  
int led = 13; //LED connected to pin 13  
  
// the setup function runs once when you press reset or power the board  
void setup() {
```

Annotations: A red arrow points from the multi-line comment in the list to the corresponding multi-line comment in the code. A blue arrow points from the single-line comment in the list to the corresponding single-line comment in the code.

COSA SONO GLI SKETCH

Variabile:

- Porzione di memoria, identificata da un'etichetta, contenente un dato; deve essere *dichiarata* prima di essere usata;
- Può contenere diversi *tipi* di dato che vanno indicati con:
 - **boolean**: può contenere due valori, **VERO** o **FALSO**;
 - **char**: può contenere un carattere, es. **'A'**;
 - **int**: può contenere un valore numerico intero compreso fra **-32'768** e **32'767**;
 - **long**: può contenere un valore numerico intero compreso fra **-2'147'483'648** e **2'147'483'647**;
 - **float**: può contenere numeri in **virgola mobile**;
 - **string**: può contenere **stringhe**.

ATTENZIONE: esistono delle parole riservate che non possono essere utilizzate come nomi per variabili e funzioni. Grazie al **syntax highlighting** è possibile riconoscere queste parole dal diverso colore con la quale vengono scritte (es. `delay`, `HIGH`, ecc).

```
modified 8 May 2014
by Scott Fitzgerald

modified 2 Sep 2016
by Arturo Guadalupi

modified 8 Sep 2016
by Colby Newman
*/
int led = 13; //LED connected to pin 13

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

// the loop function 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
}
```

Compilazione completata

Lo sketch usa 940 byte (2%) dello spazio disponibile per i programmi. Le variabili globali usano 9 byte (0%) di memoria dinamica, lasciando ancora 1006 byte liberi.

COSA SONO GLI SKETCH

Istruzione:

- Riga di codice che può contenere una dichiarazione, un'assegnazione o la *chiamata* a una funzione;
- Ogni istruzione deve terminare necessariamente con ;.

```
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi

modified 3 Sep 2016
by Colby Newman
*/

int led; //LED connected to pin 13
// the setup function runs once when you press reset or power
void setup() {
  led = 13;
  // initialize digital pin "led" as an output.
  pinMode(led, OUTPUT);
}

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

Completazione completata

Lo sketch usa 953 byte (2%) dello spazio disponibile per i prog
Le variabili globali usano 11 byte (0%) di memoria dinamica, 12

1

COSA SONO GLI SKETCH

Strutture di controllo:

- Le strutture di controllo sono costrutti che servono a specificare **se, quando, in quale ordine e quante volte** devono essere eseguite le istruzioni che compongono il codice sorgente.

```
int tasto = 0;
void loop() {
  if (tasto == 0)
  {
    //tasto non premuto
  }
  else
  {
    //tasto premuto
  }
}
```

costrutto
if - else

```
int numero;
void loop() {
  for (numero = 0; numero < 10; numero = numero + 1)
  {
    //questa istruzione viene ripetuta per 10 volte
  }
}
```

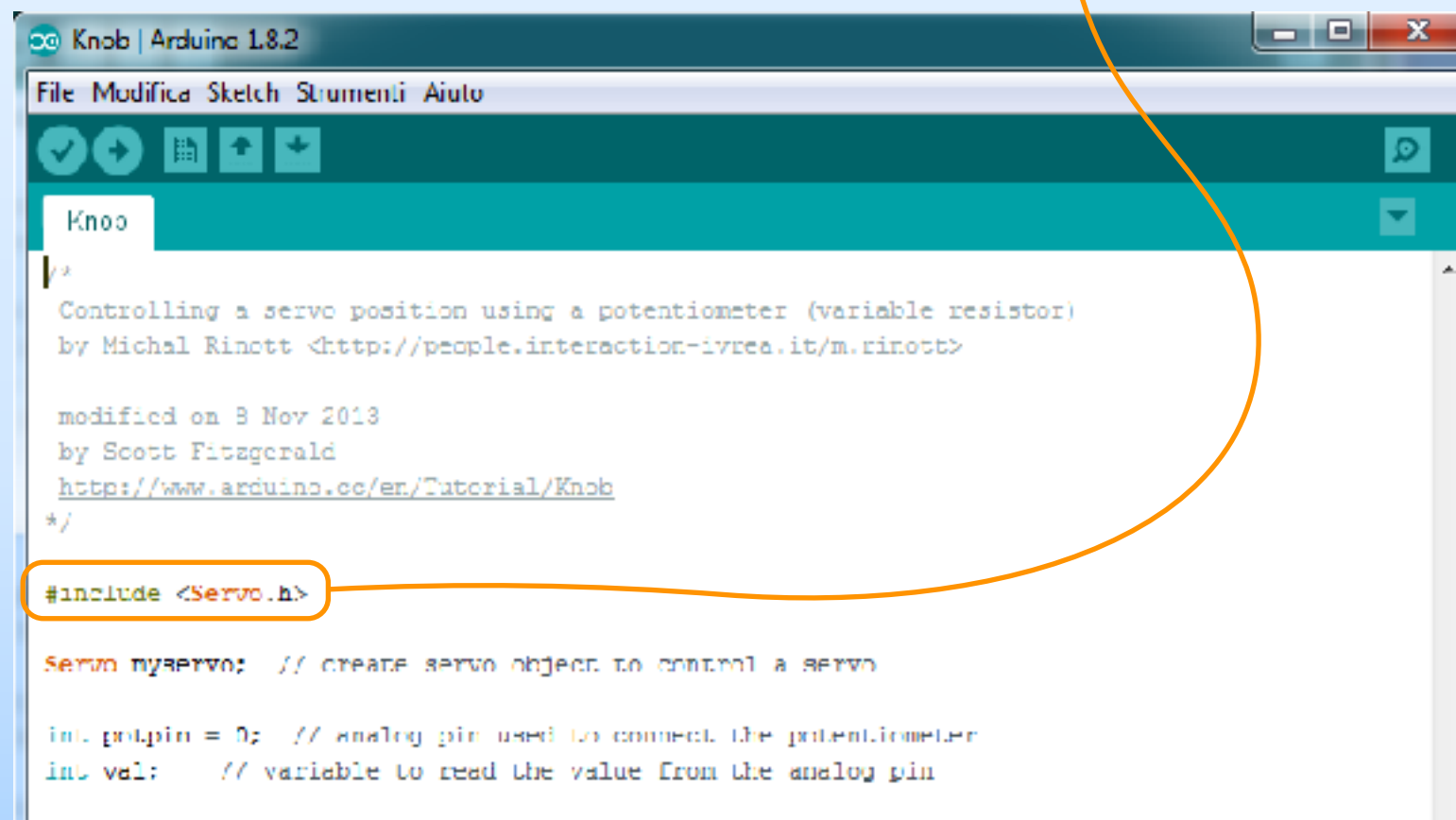
costrutto
for



COSA SONO GLI SKETCH

Libreria:

- È un insieme di funzioni o dati, contenuto in uno o più file e che può essere *incluso* all'interno di un programma attraverso un collegamento;
- L'inclusione all'interno dello sketch avviene attraverso la direttiva **#include**.



```
Knob | Arduino 1.8.2
File Modifica Sketch Strumenti Aiuto
Knob
/*
Controlling a servo position using a potentiometer (variable resistor)
by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>

modified on 8 Nov 2013
by Scott Fitzgerald
http://www.arduino.cc/en/Tutorial/Knob
*/
#include <Servo.h>

Servo myservo;  // create servo object to control a servo

int potpin = 0;  // analog pin used to connect the potentiometer
int val;        // variable to read the value from the analog pin
```

COSA SONO GLI SKETCH

Principali funzioni (reference):

- **pinMode** (*pin*, ^{INPUT}_{OUTPUT}) ; imposta un pin come input o come output.
- **digitalRead** (*pin*) ; legge lo stato digitale del pin.
- **digitalWrite** (*pin*, ^{HIGH}_{LOW}) ; scrive lo stato digitale di un pin.
- **analogRead** (*pin*) ; legge il valore di un ingresso analogico.
- **analogWrite** (*pin*, *val*) ; scrive un valore analogico su un pin.
- **delay** (*ms*) ; inserisce una pausa di *ms* millisecondi.
- **Serial.begin** (*speed*) ; inizializza la seriale alla velocità di *speed* bps.
- **Serial.print** ("text") ; stampa la parola "**text**" sulla seriale.



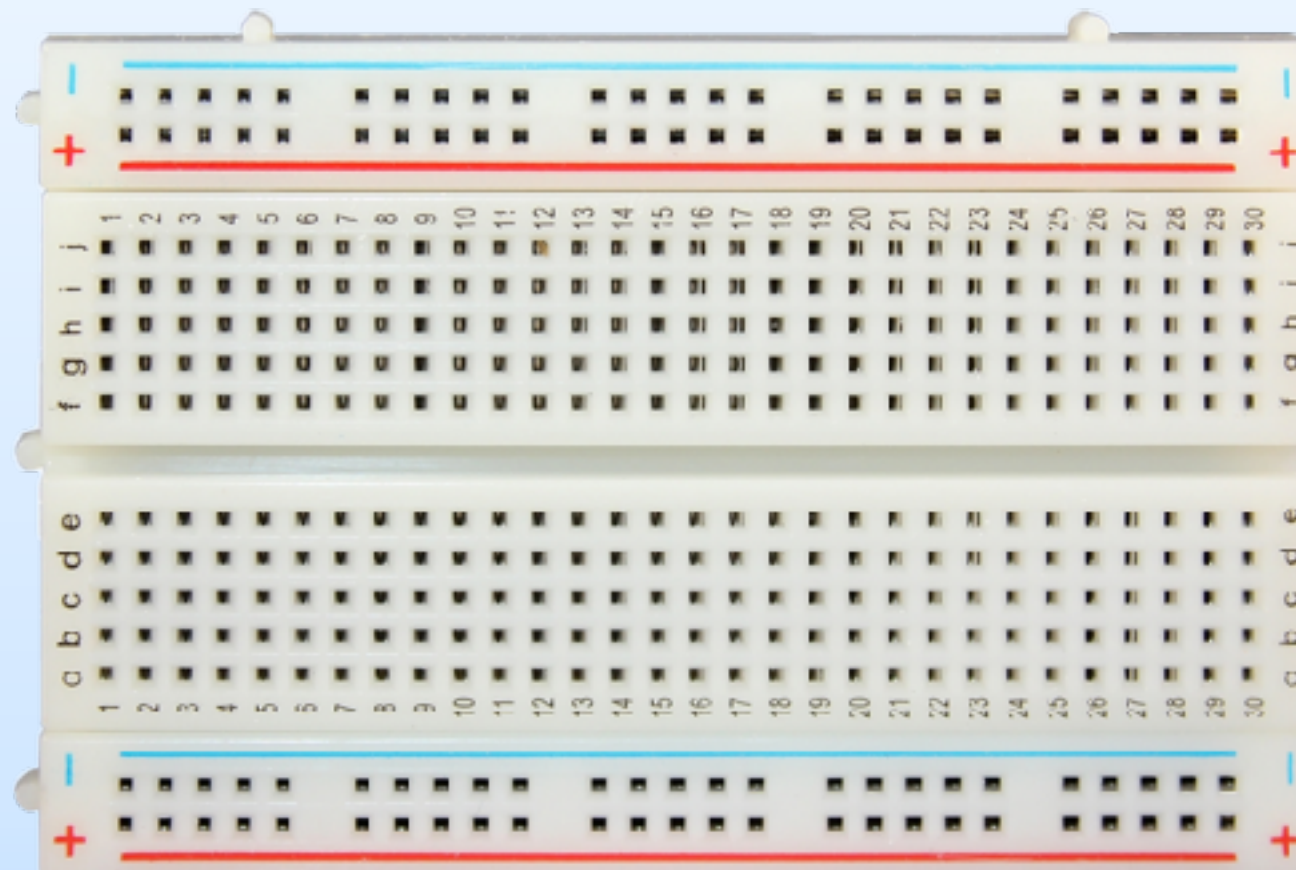
COMPONENTI DEL KIT

Componenti:

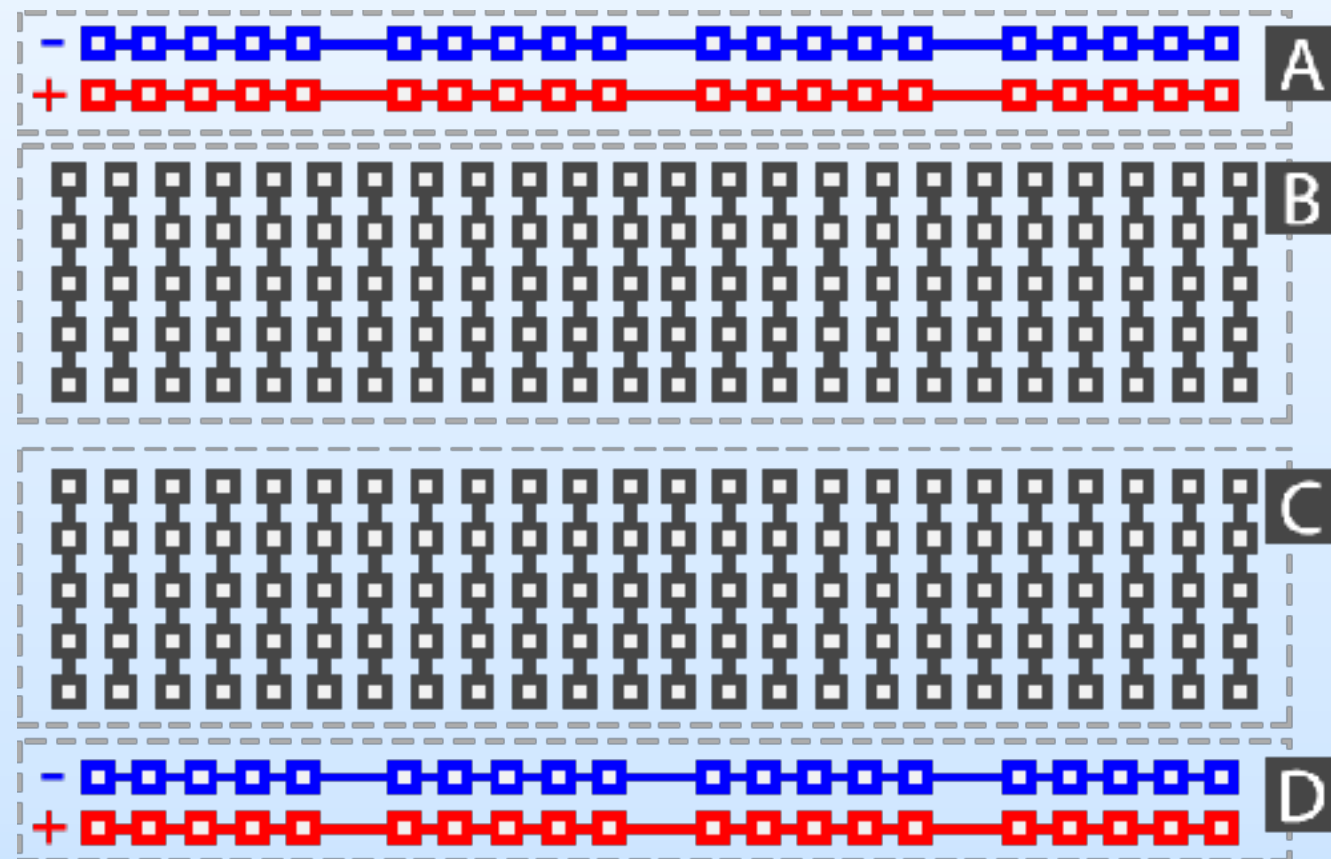
- 1 x Breadboard;
- 1 x Servo;
- 1 x DHT11;
- 1 x Potenzziometro;
- 1 x Fotorresistenza;
- 2 x Bottone;
- 2 x LED rosso;
- 2 x LED verde;
- 2 x LED giallo;
- 1 x LED RGB;
- 10 x Resistenza 10K Ohm;
- 10 x Resistenza 220 Ohm;
- 10 x jumper M-M.



Componenti del KIT BREADBOARD



Componenti del KIT BREADBOARD



BLINK

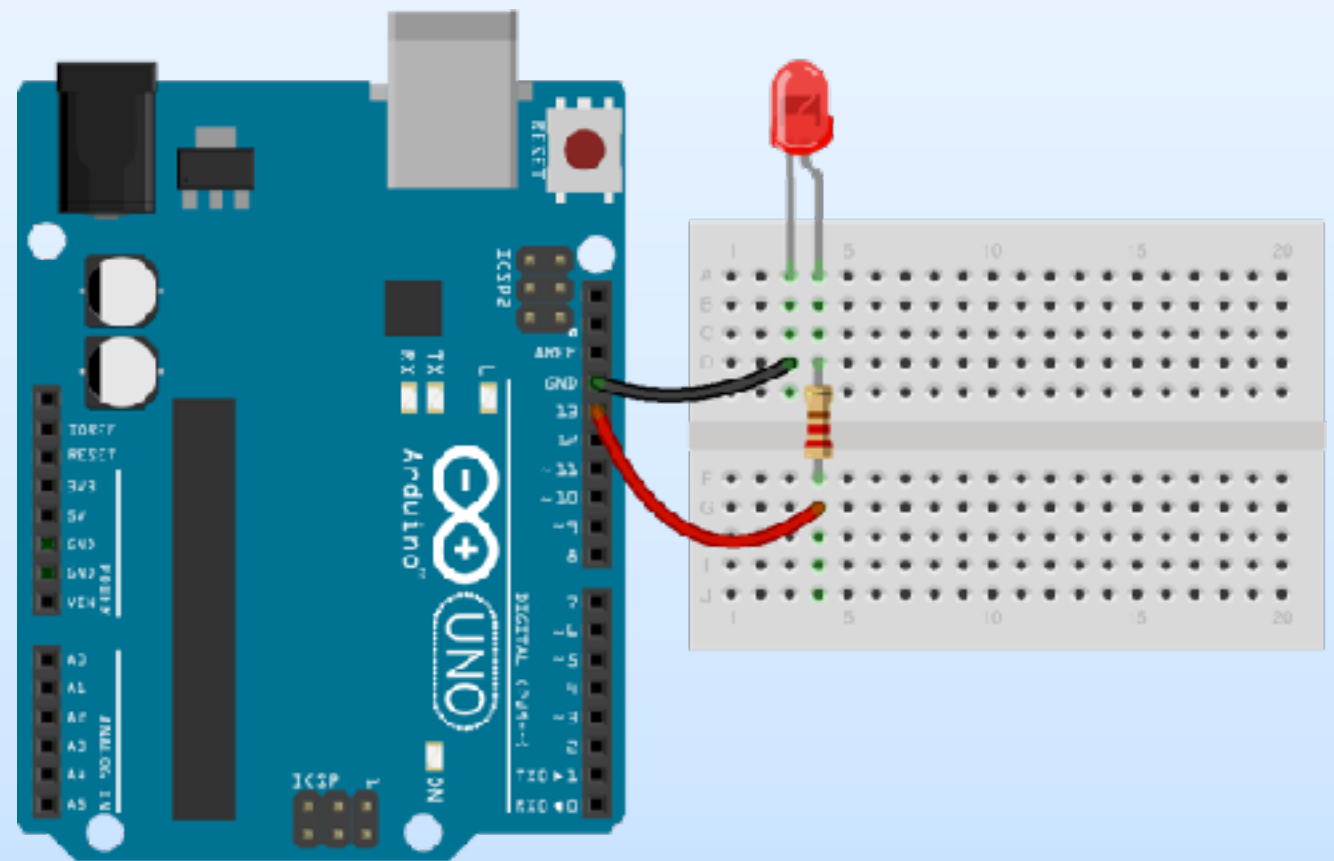
Componenti:

- 1 x Breadboard
- 1 x Arduino;
- 1 x resistenza 220Ω ;
- 1 x LED;

Sketch:

```
int led = 13;
void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```



BUTTON

Componenti:

- 1 x Breadboard;
- 1 x Arduino;
- 1 x resistenza 220 Ω ;
- 1 x LED;
- 1 x bottone;
- 1 x resistenza 10 K Ω .

Sketch:

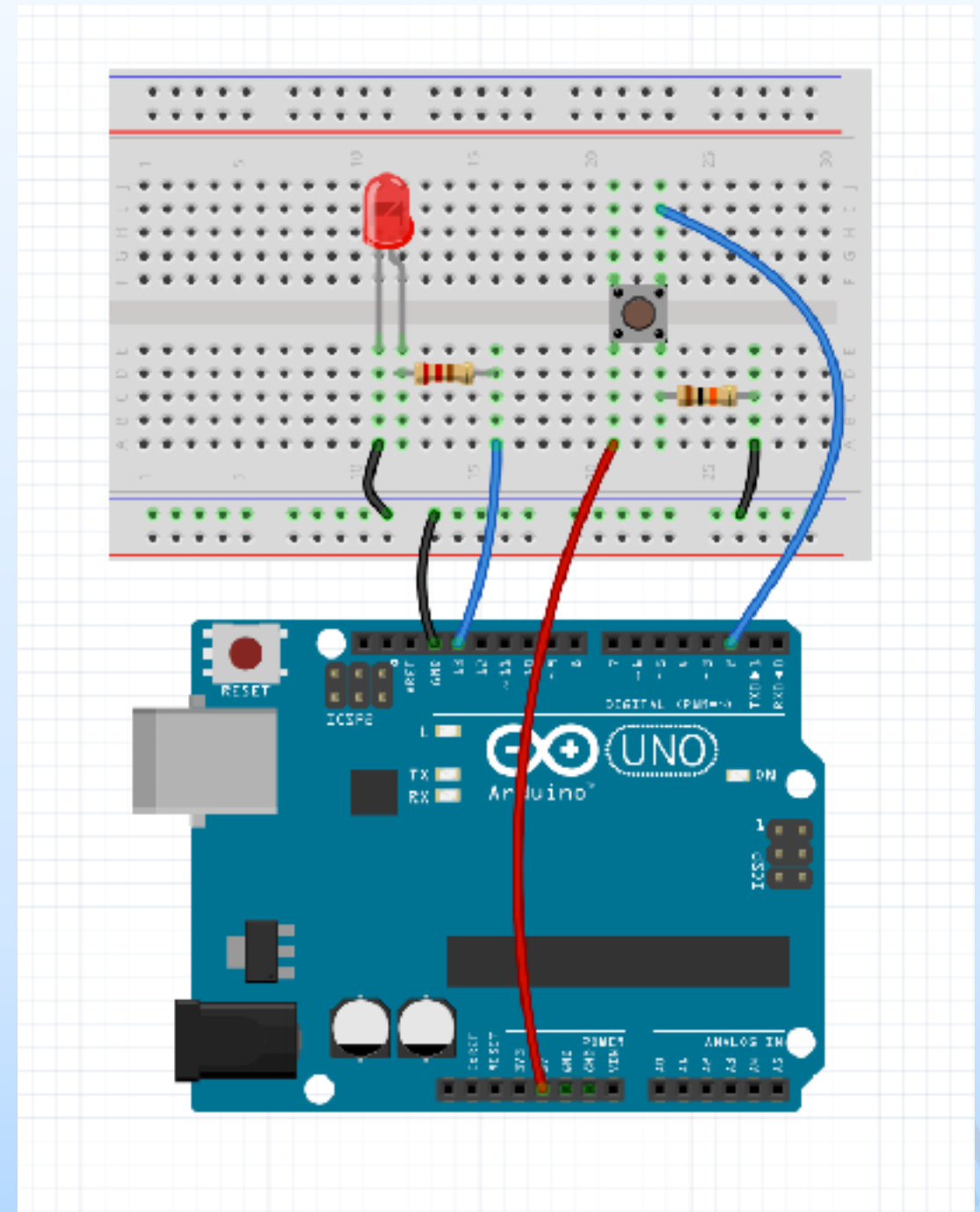
```
int pinBottone = 2;
int pinLed = 13;

int statoBottone = 0;

void setup() {
  pinMode(pinLed, OUTPUT);
  pinMode(pinBottone, INPUT);
}

void loop() {
  statoBottone = digitalRead(pinBottone);

  if (statoBottone == HIGH) {
    digitalWrite(pinLed, HIGH);
  }
  else
  {
    digitalWrite(pinLed, LOW);
  }
}
```



ANALOG INPUT - ANALOG OUTPUT

Componenti:

- 1 x Breadboard;
- 1 x Arduino;
- 1 x resistenza 220 Ω ;
- 1 x LED;
- 1 x potenziometro.

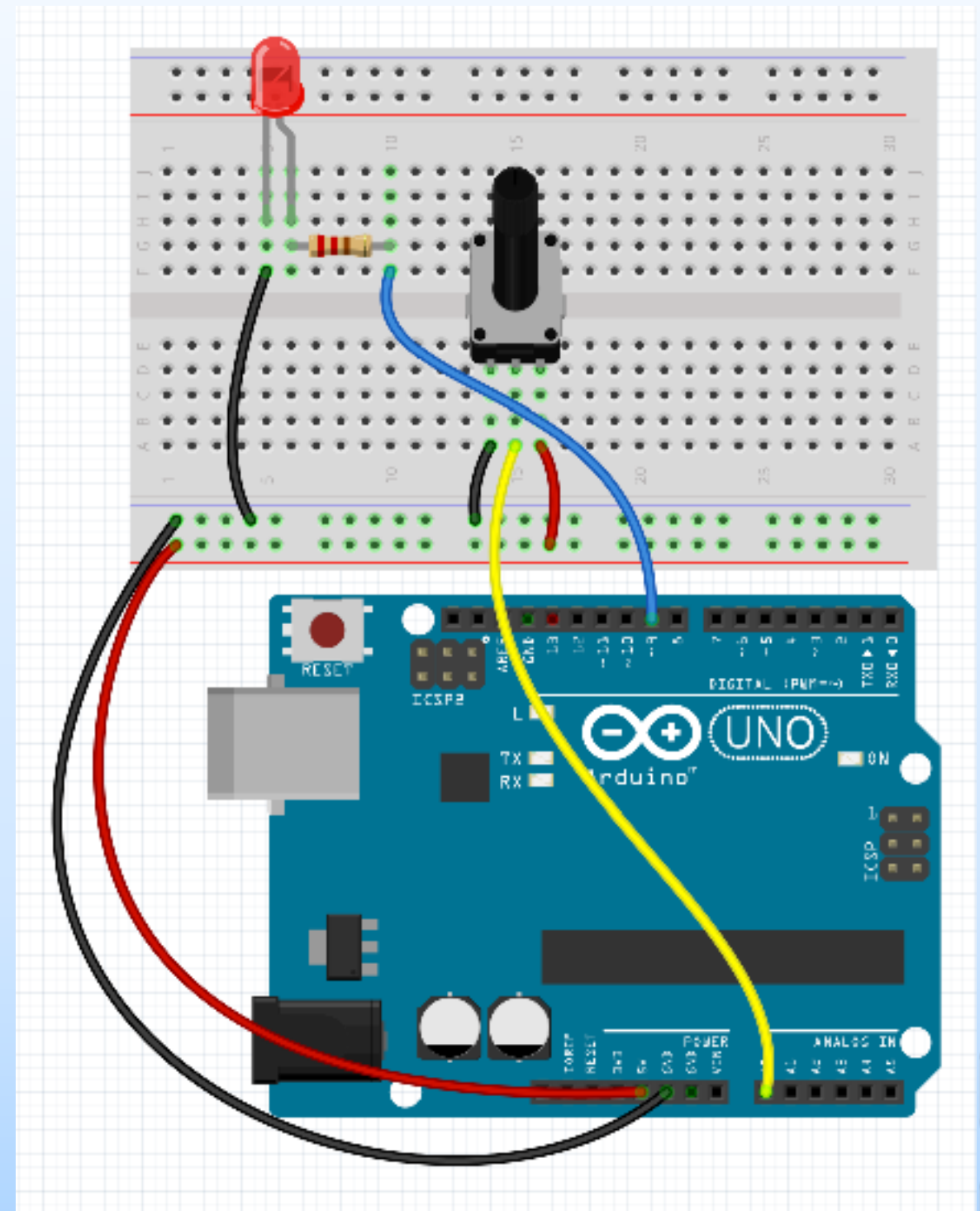
Sketch:

```
int pinPot = A0;
int pinLed = 9;
int valoreSensore = 0;

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

void loop() {
  valoreSensore = analogRead(pinPot);
  valoreSensore = map(valoreSensore, 0, 1023, 0, 255);

  analogWrite(pinLed, valoreSensore);
}
```



ON-OFF LED DA SERIALE

Componenti:

- 1 x Breadboard
- 1 x Arduino;
- 1 x resistenza 220 Ω ;
- 1 x LED.

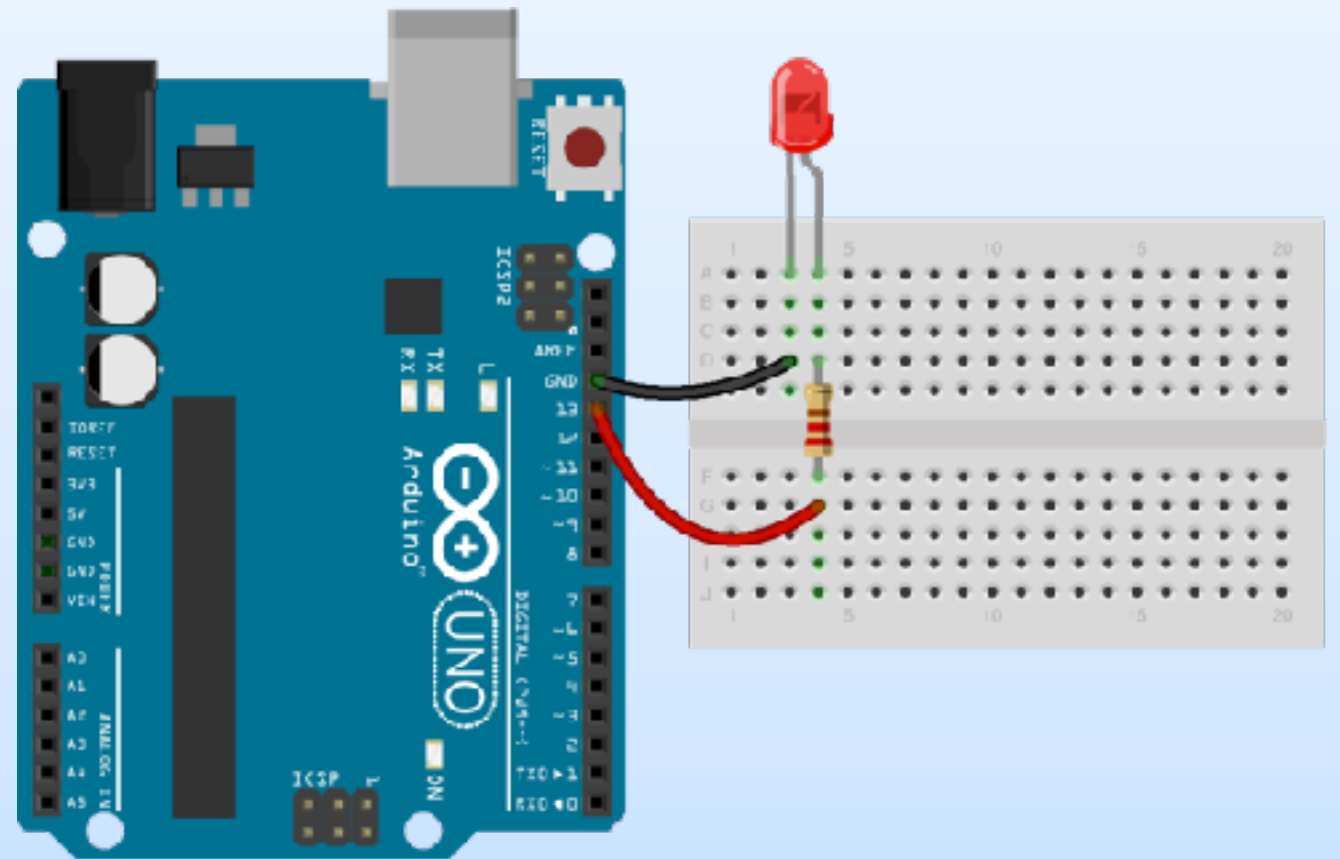
Sketch:

```
int led = 13;
char carattere;

void setup() {
  Serial.begin(9600);
  pinMode(led, OUTPUT);
}

void loop() {
  if (Serial.available() > 0) {
    carattere = Serial.read();

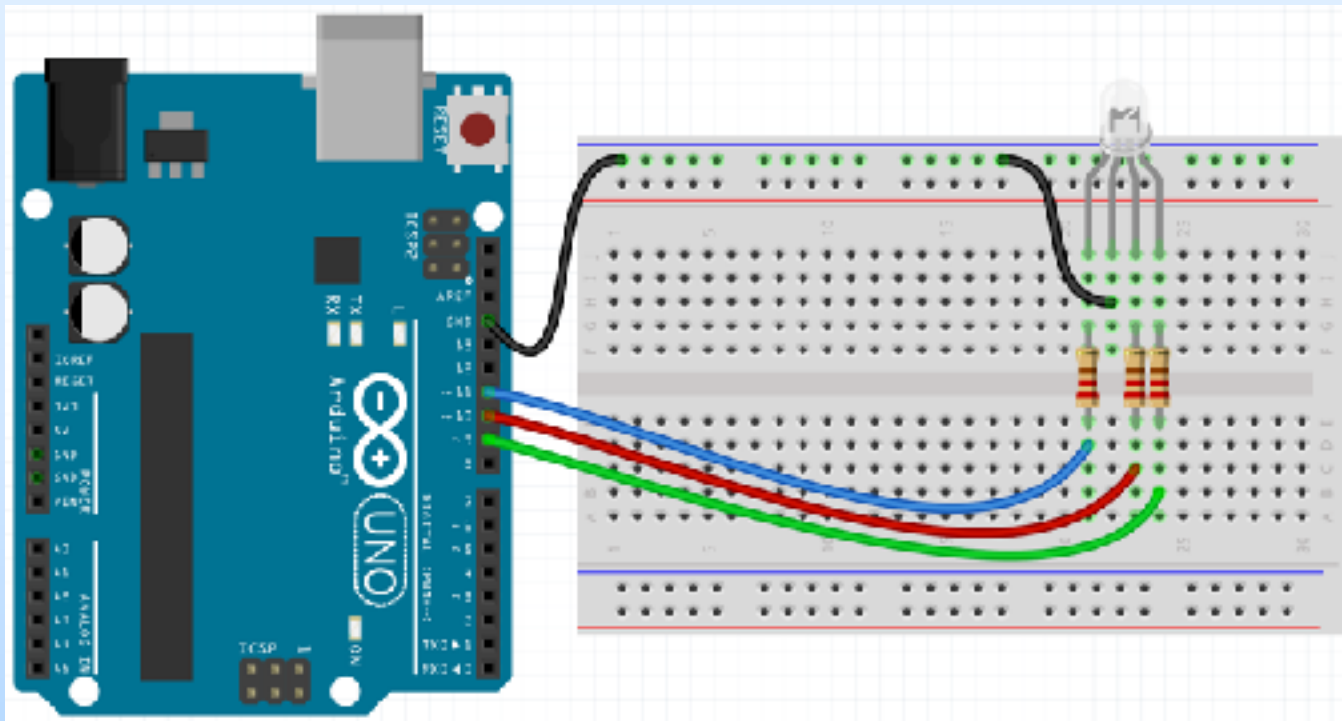
    if (carattere == 'H') {
      digitalWrite(led, HIGH);
    }
    else if (carattere == 'L') {
      digitalWrite(led, LOW);
    }
  }
}
```



SEMAFORO LED RGB

Componenti:

- 1 x Breadboard
- 1 x Arduino;
- 3 x resistenza 220 Ω ;
- 1 x LED RGB.



Sketch:

```
int ledR = 10;
int ledG = 9;
int ledB = 11;

void semaforo(int colore);

void setup() {
  pinMode(ledR, OUTPUT);
  pinMode(ledG, OUTPUT);
  pinMode(ledB, OUTPUT);
}

void loop() {
  semaforo(ledR);
  delay(3000);

  semaforo(ledG);
  delay(3000);

  semaforo(ledB);
  delay(3000);
}

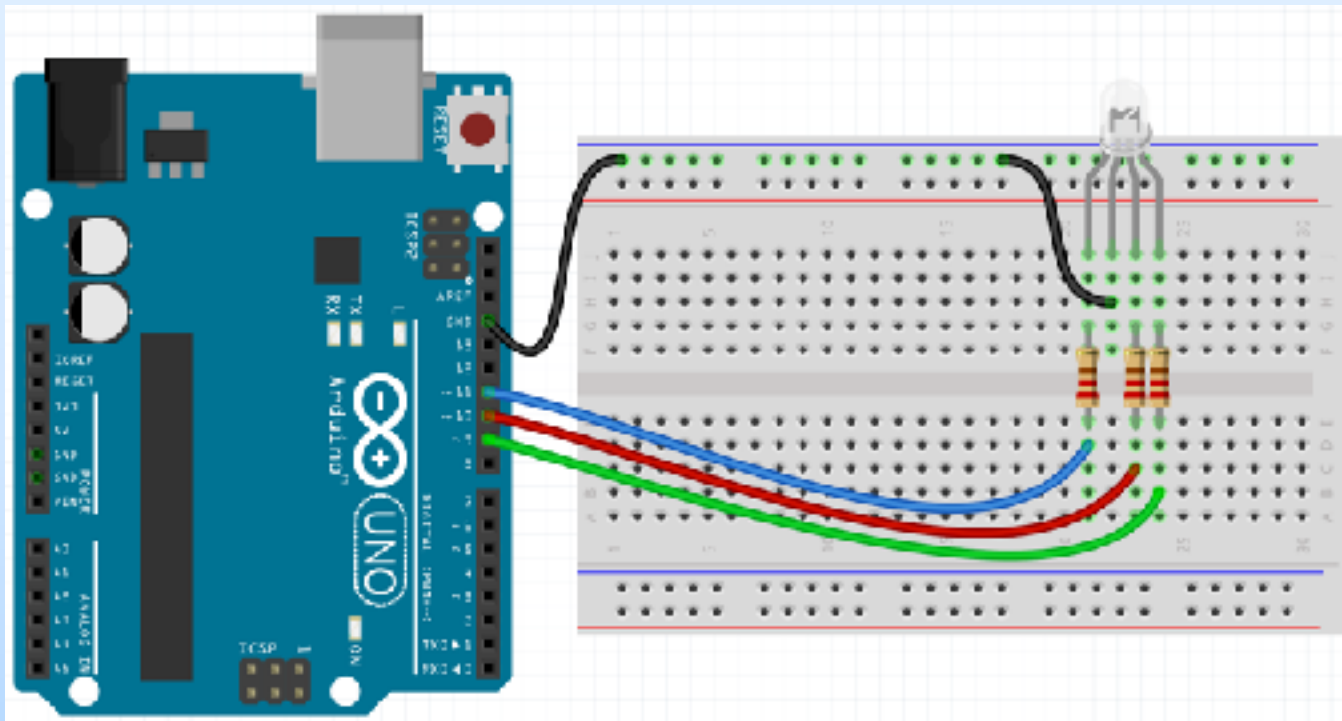
void semaforo(int colore) {
  digitalWrite(ledR, LOW);
  digitalWrite(ledG, LOW);
  digitalWrite(ledB, LOW);

  digitalWrite(colore, HIGH);
}
```


ON-OFF LED RGB DA SERIALE

Componenti:

- 1 x Breadboard
- 1 x Arduino;
- 3 x resistenza 220 Ω ;
- 1 x LED RGB.



Sketch:

```
int ledR = 10;
int ledG = 9;
int ledB = 11;
char carattere;

void setup() {
  Serial.begin(9600);
  pinMode(ledR, OUTPUT);
  pinMode(ledG, OUTPUT);
  pinMode(ledB, OUTPUT);
}

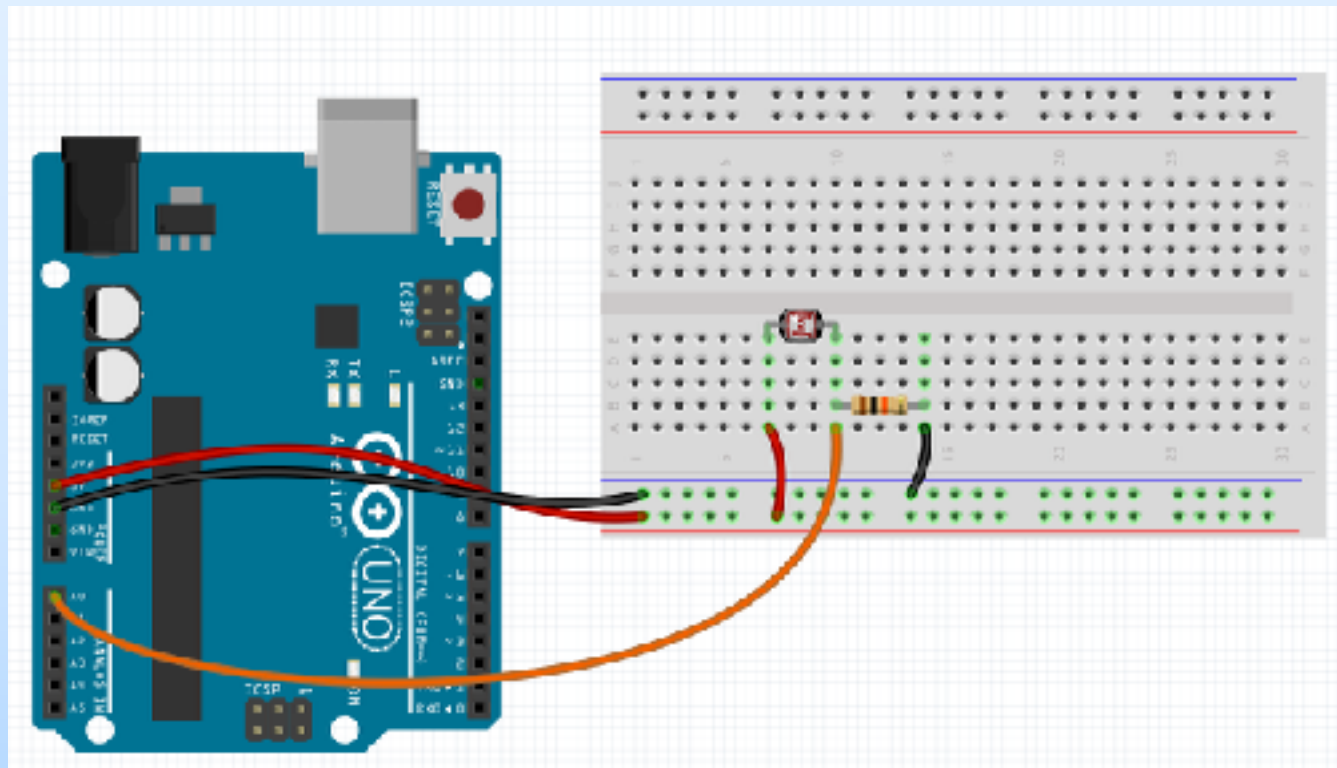
void loop() {
  if (Serial.available() > 0) {
    carattere = Serial.read();

    if (carattere == 'R') {
      digitalWrite(ledR, HIGH);
    }
    else if (carattere == 'r') {
      digitalWrite(ledR, LOW);
    }
    else if (carattere == 'G') {
      digitalWrite(ledG, HIGH);
    }
    else if (carattere == 'g') {
      digitalWrite(ledG, LOW);
    }
    else if (carattere == 'B') {
      digitalWrite(ledB, HIGH);
    }
    else if (carattere == 'b') {
      digitalWrite(ledB, LOW);
    }
  }
}
```


FOTORESISTENZA

Componenti:

- 1 x Breadboard
- 1 x Arduino;
- 1 x fotoresistenza;
- 1 x resistenza 10 K Ω .



Sketch:

```
int led = 13;
int pinFoto = A0;
int valore = 0;

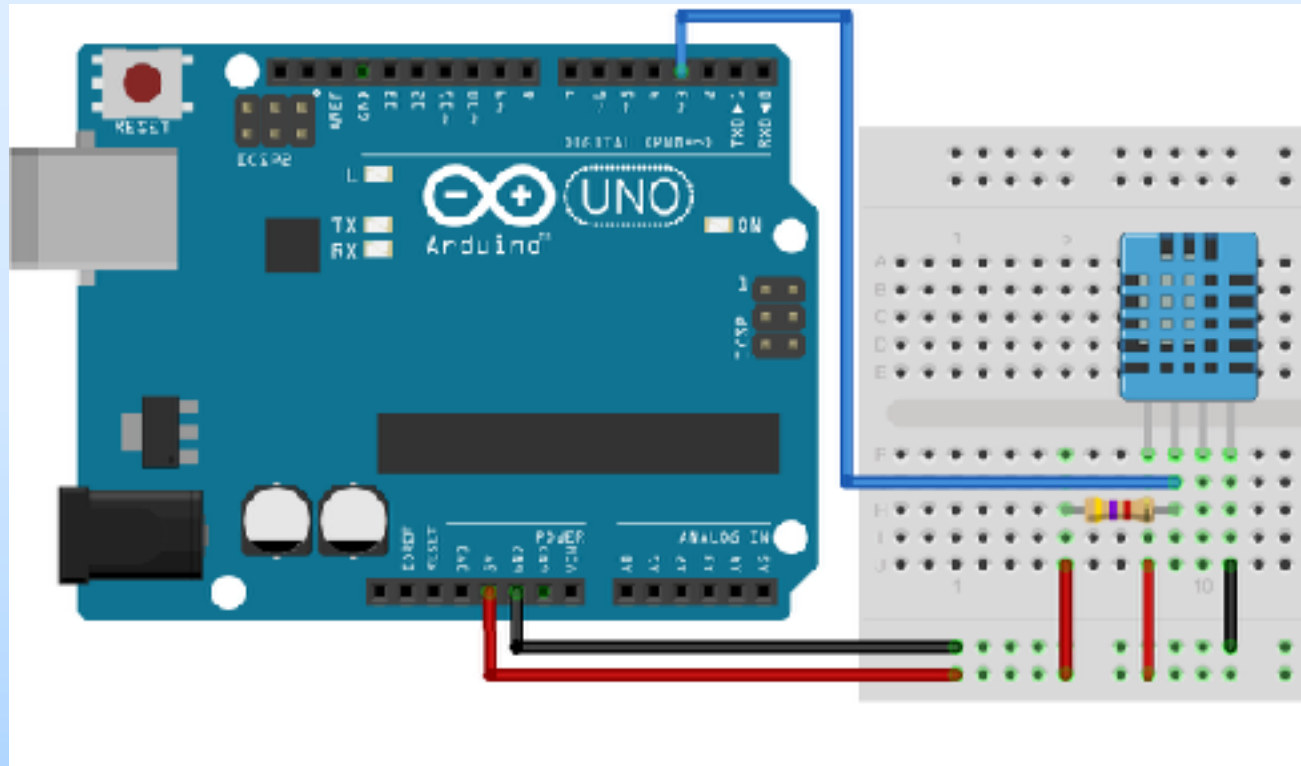
void setup() {
  Serial.begin(9600);
  pinMode(led, OUTPUT);
}

void loop() {
  valore = analogRead(pinFoto);
  Serial.print("Valore letto = ");
  Serial.println(valore);
}
```

SENSORE DHT11

Componenti:

- 1 x Breadboard
- 1 x Arduino;
- 1 x sensore DHT11;
- 1 x resistenza 5.7 K Ω .



Sketch:

[DOWNLOAD LIBRERIA DHT](#)

```
#include "DHT.h"    //includiamo la libreria
int pinDHT = 3;

DHT dht(pinDHT, DHT11);

void setup() {
  Serial.begin(9600);
  Serial.println("Test temperatura e umidità!");
  Serial.println("T(C)  \tH(%)");
  dht.begin();
}

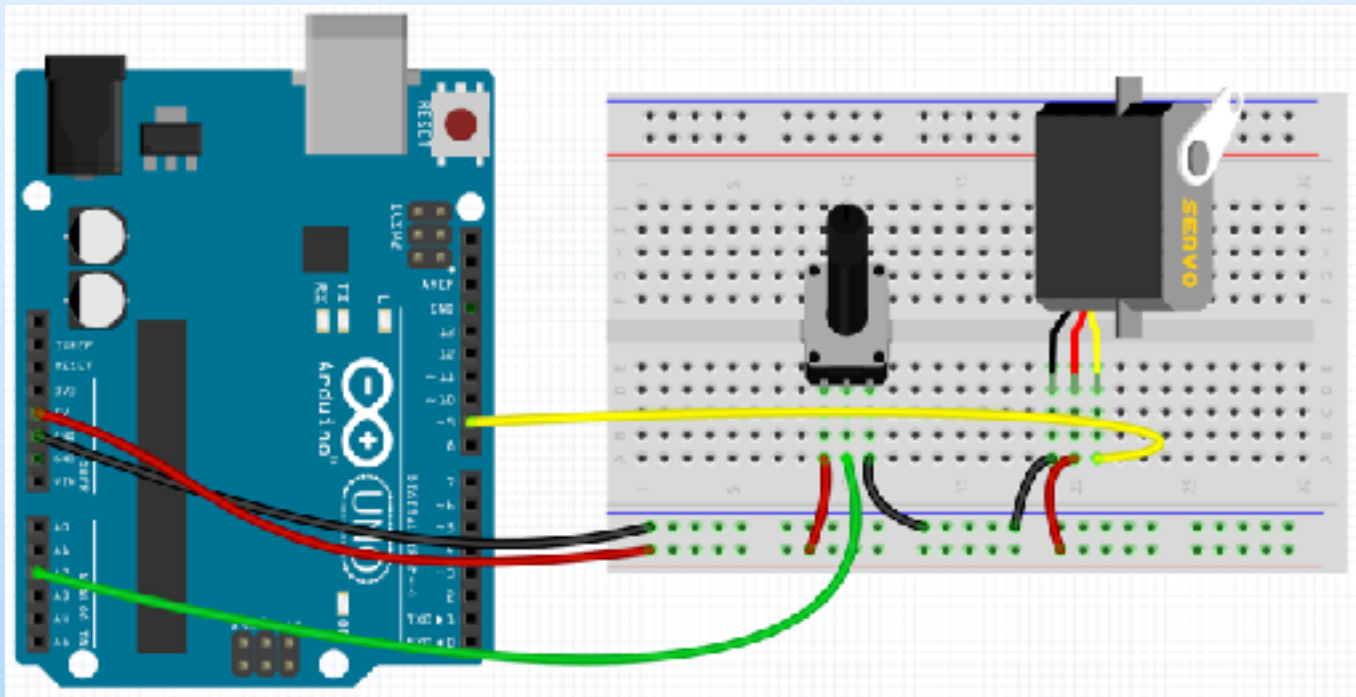
void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println("Lettura fallita!");
  }
  else
  {
    Serial.print(t, 2);    //2 cifre decimali
    Serial.print("\t");
    Serial.println(h, 2);  //2 cifre decimali
    delay(2000);
  }
}
```

SERVO

Componenti:

- 1 x Breadboard
- 1 x Arduino;
- 1 x Servo;
- 1 x potenziometro;



Sketch:

```
#include <Servo.h>    //includiamo la libreria
int pinPot = A3;
int pinServo = 9;
int val;

Servo myservo;

void setup() {
  myservo.attach(pinServo);
}

void loop() {
  val = analogRead(pinPot);
  val = map(val, 0, 1023, 0, 180);
  myservo.write(val);
  delay(15);
}
```