

# Led e LCD in Arduino

Programmazione di sistemi multicore

Michele Martinelli

[Michele.martinelli@uniroma1.it](mailto:Michele.martinelli@uniroma1.it)

Marco Bernardi

[m.bernardi@uniroma1.it](mailto:m.bernardi@uniroma1.it)

Georgia Koutsandria

[georgia.koutsandria@uniroma1.it](mailto:georgia.koutsandria@uniroma1.it)

# Avvisi

- **Giovedì 24 Ottobre 2019 NON CI SARA LEZIONE!!!**
  - **Controllare twiki per conoscere quando sarà recuperata**
- Per la lezione di Lunedì 28 installare Processing:
  - <https://processing.org/download/>

# Comunicazione Seriale in Arduino

- **Serial.begin(speed)** – Imposta la velocità dei dati in bit al secondo (baud) per la trasmissione seriale dei dati

Ex: `Serial.begin(9600);` // imposta la comunicazione a 9600 bit per secondo

- **Serial.println(sensorValue)** – Stampa in output (sul monitor seriale) il valore *sensorValue*

Ex: `value = analogRead(A0);`  
`Serial.println(value);`

- **Serial.available()** verifica se ci sono dati da leggere nel buffer della seriale
- **Serial.read()** legge i dati da seriale
- **Serial.flush()** attende il completamento della trasmissione dei dati seriali in uscita.

# Responder Experiment

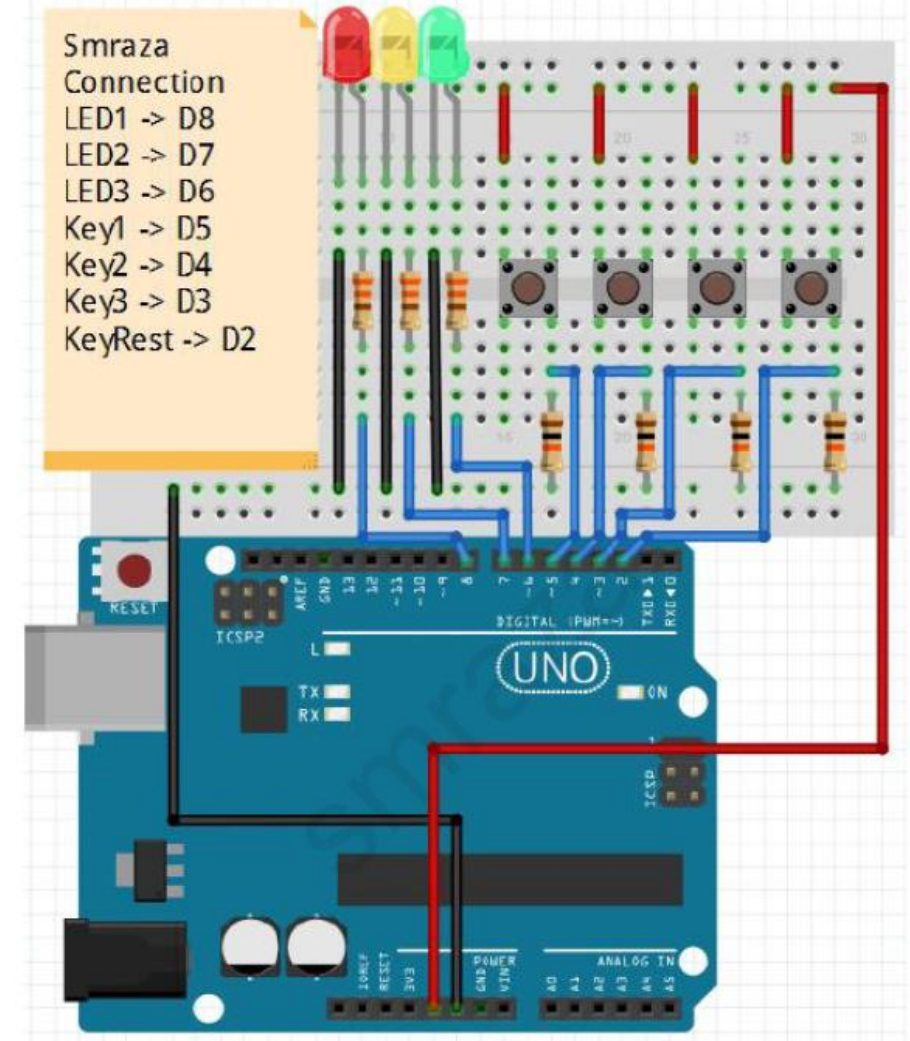
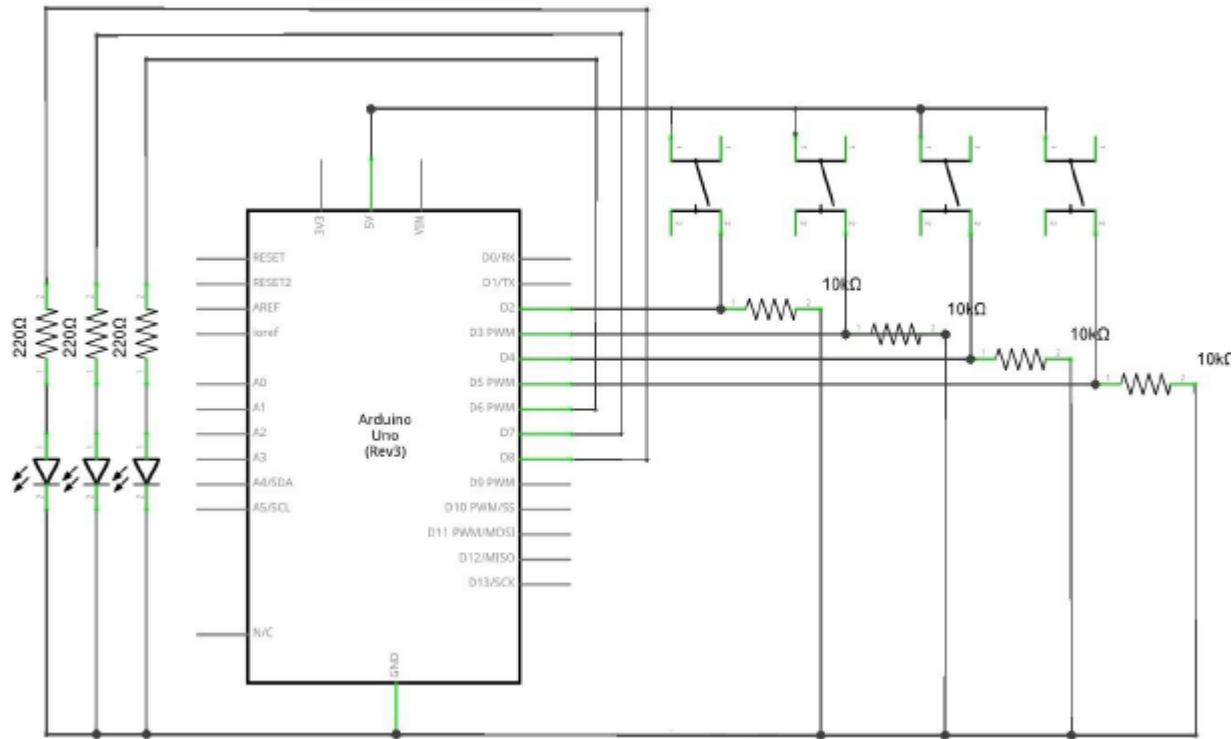


In questo esercizio vedremo l'utilizzo di come si possa utilizzare più bottoni per controllare l'illuminazione di più LED.

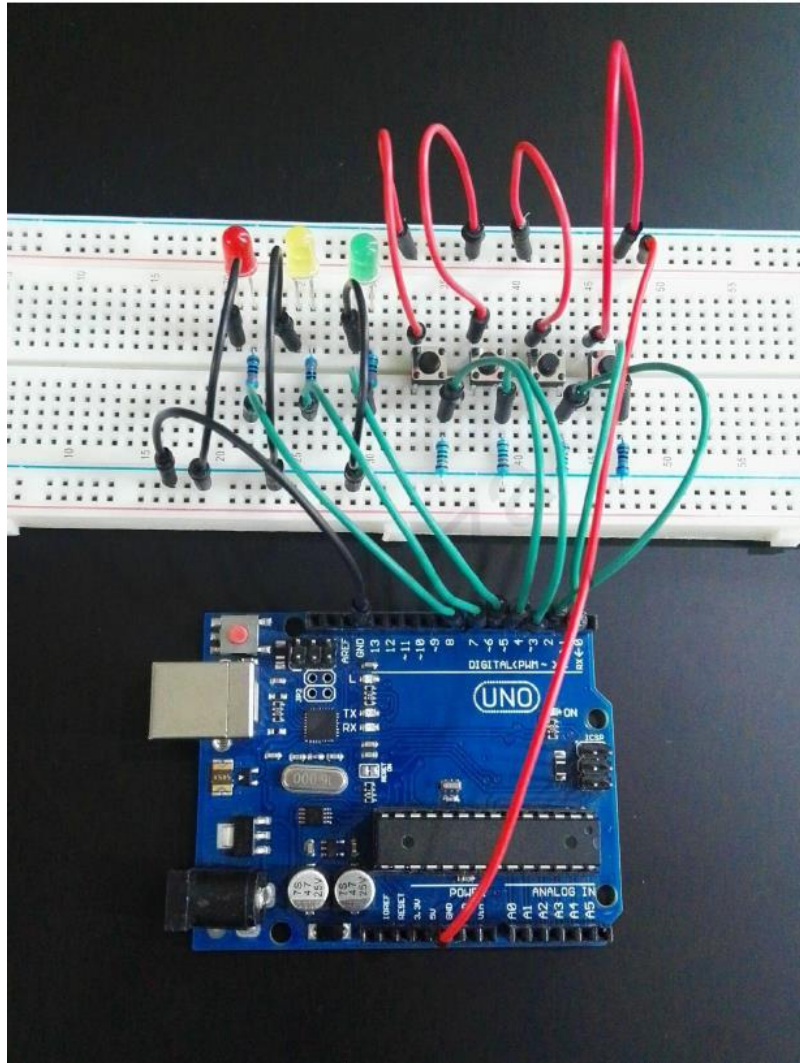
## Hardware Richiesto:

- 3 LED
- 4 Switch o bottoni
- 3 Resistenze da 220/330  $\Omega$
- 4 Resistenze da 10 K $\Omega$
- BreadBoard e cavi
- Arduino
- Cavo USB

# Responder Experiment: Costruire il circuito



# Il programma in Arduino



```
int Redled=8;
int Yellowled=7;
int Greenled=6;
int Key1=5;
int Key2=4;
int Key3=3;
int KeyRest=2;
int Red;
int Yellow;
int Green;

void setup()
{
  pinMode(Redled,OUTPUT);
  pinMode(Yellowled,OUTPUT);
  pinMode(Greenled,OUTPUT);
  pinMode(Key1,INPUT);
  pinMode(Key2,INPUT);
  pinMode(Key3,INPUT);
  pinMode(KeyRest,INPUT);
}

void loop()
{
  Red=digitalRead(Key1);
  Yellow=digitalRead(Key2);
  Green=digitalRead(Key3);
  if(Red==HIGH) Red_YES();
  if(Yellow==HIGH) Yellow_YES();
  if(Green==HIGH) Green_YES();
}

void Red_YES()
{
  while(digitalRead(KeyRest)==0)
  {
    digitalWrite(Redled,HIGH);
    digitalWrite(Greenled,LOW);
    digitalWrite(Yellowled,LOW);
  }
  clear_led();
}

void Yellow_YES()
{
  while(digitalRead(KeyRest)==0)
  {
    digitalWrite(Redled,LOW);
    digitalWrite(Greenled,LOW);
    digitalWrite(Yellowled,HIGH);
  }
  clear_led();
}

void Green_YES()
{
  while(digitalRead(KeyRest)==0)
  {
    digitalWrite(Redled,LOW);
    digitalWrite(Greenled,HIGH);
    digitalWrite(Yellowled,LOW);
  }
  clear_led();
}

void clear_led()
{
  digitalWrite(Redled,LOW);
  digitalWrite(Greenled,LOW);
  digitalWrite(Yellowled,LOW);
}
```

# Responder Experiment



In questo esercizio impareremo come utilizzare un LED RGB (rosso verde blu) con un Arduino.

Si utilizzerà la funzione `analogWrite` per controllare il colore del LED.

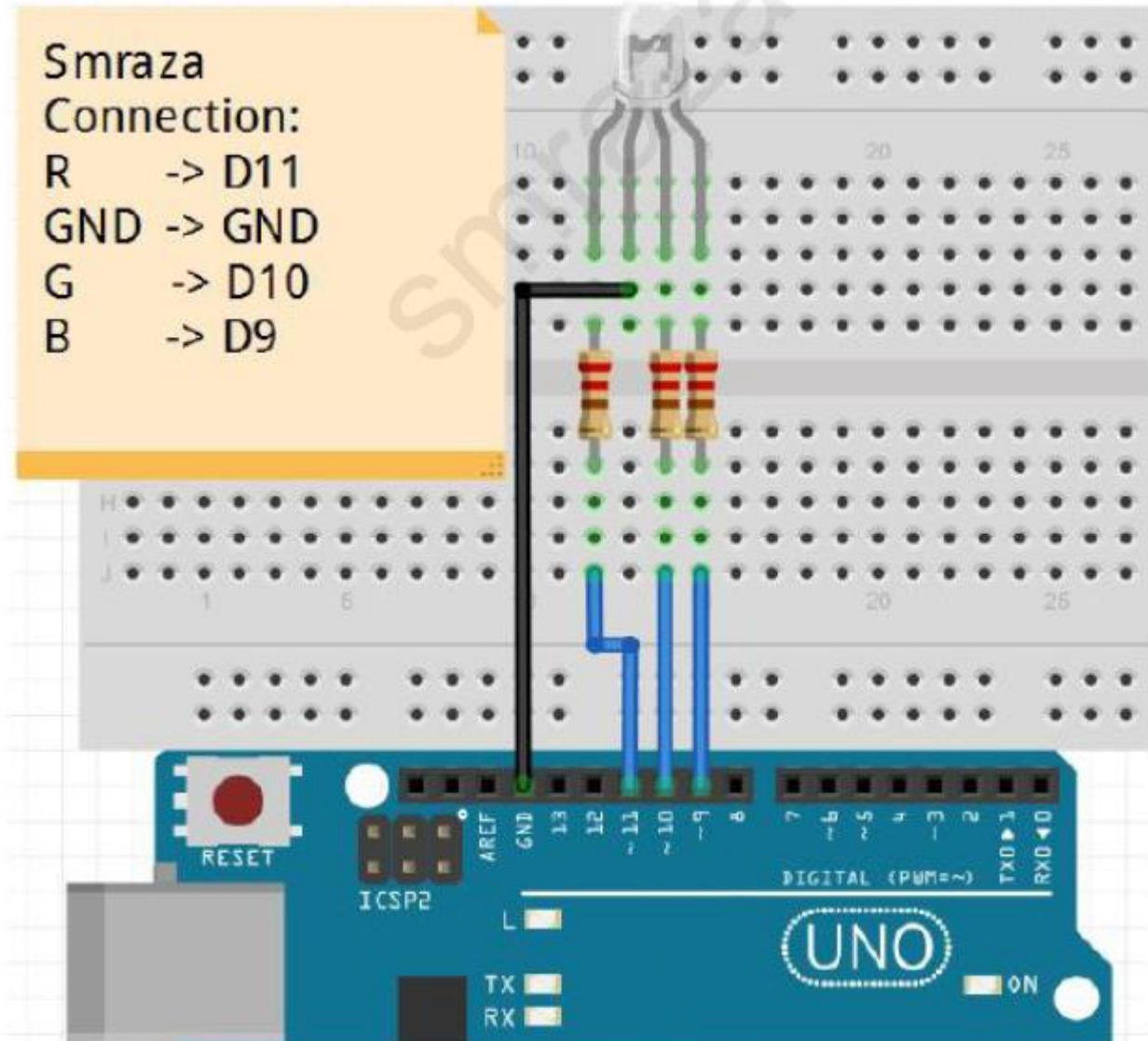
## Hardware Richiesto:

- 1 LED RGB
- 3 Resistenze da 220/330  $\Omega$
- BreadBoard e cavi
- Arduino
- Cavo USB

# Responder Experiment: Costruire il circuito

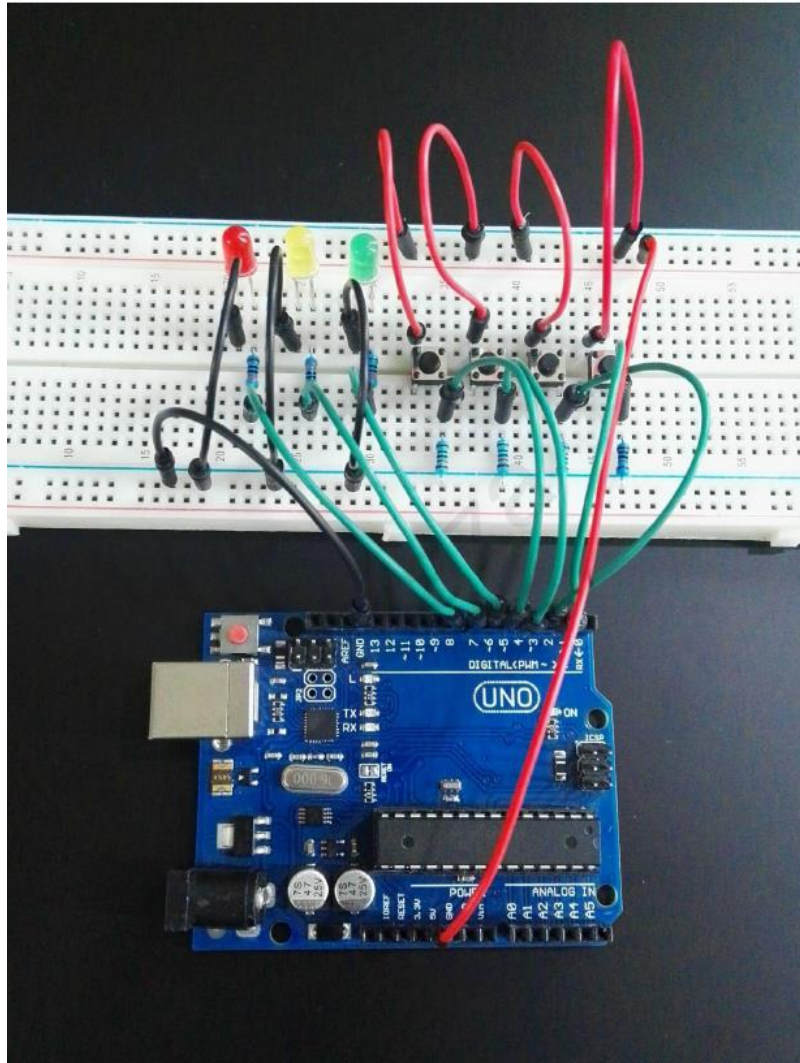


Il pin più lungo del LED RGB è collegato al Ground. Il resto dei pin, che rappresenta un singolo canale Del LED, è collegato come se fosse un LED a sé Stante.





# Il programma in Arduino



```
int redPin = 11;
int greenPin = 10;
int bluePin = 9;

void setup()
{
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}
void loop()
{
  setColor(255, 0, 0); // rosso
  delay(1000);
  setColor(0, 255, 0); // verde
  delay(1000);
  setColor(0, 0, 255); // blu
  delay(1000);
  setColor(255, 255, 0); // giallo
  delay(1000);
  setColor(80, 0, 80); // viola
  delay(1000);
  setColor(0, 255, 255); // acqua
  delay(1000);
}
void setColor(int red, int green, int blue)
{
  #ifdef COMMON_ANODE
  red = 255 - red;
  green = 255 - green;
  blue = 255 - blue;
  #endif
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}
```

# LCD



In questo esercizio impareremo come utilizzare un LCD con un Arduino. Si utilizzerà la libreria LiquidCrystal per controllare il colore del LED.

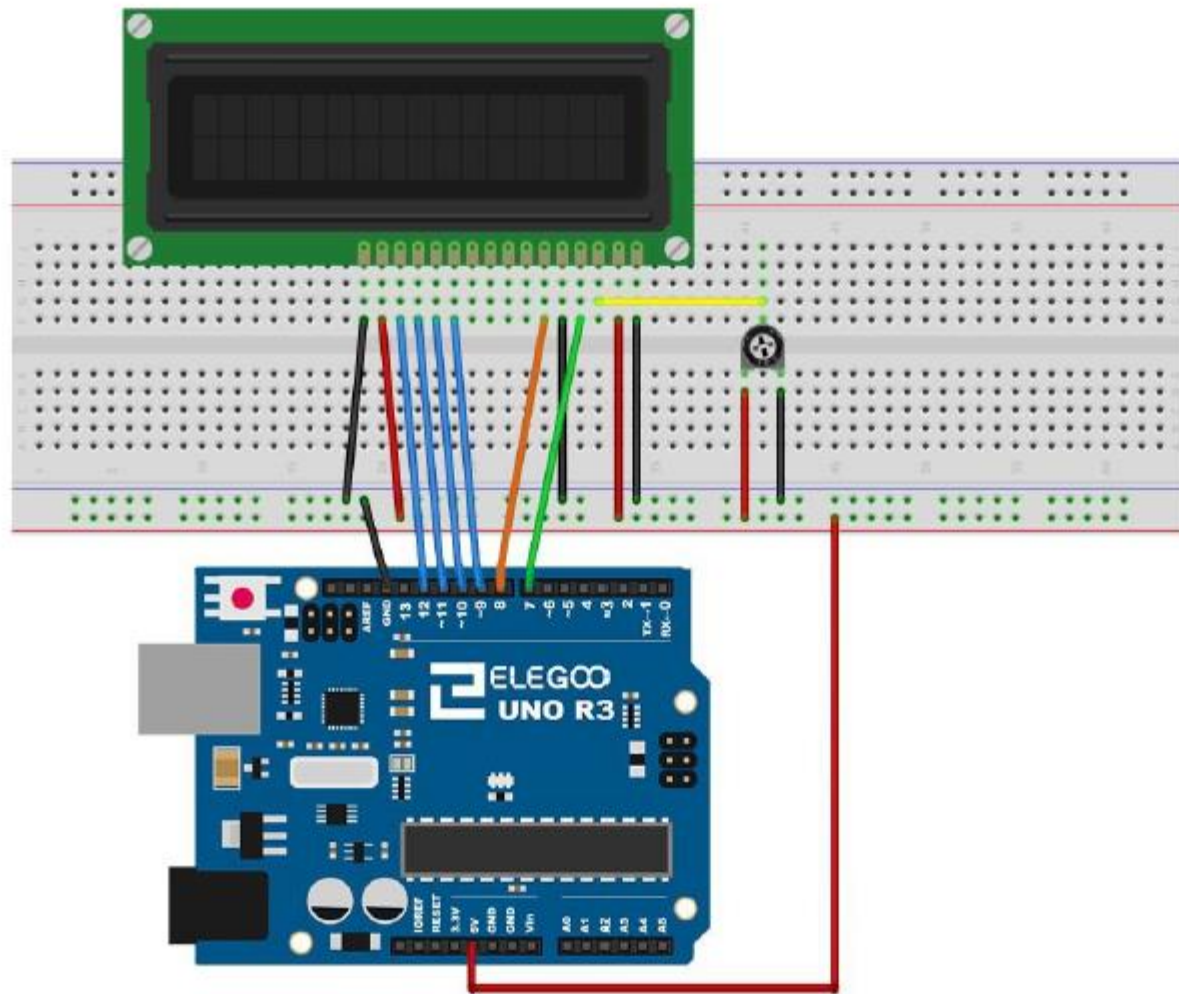
## Hardware Richiesto:

- 1 LCD
- 1 Potenzimetro 10K $\Omega$
- BreadBoard e cavi
- Arduino
- Cavo USB

# Liquid Crystal <LiquidCrystal.h>

- **Begin(row,col)** – Imposta il numero di righe e colonne del LCD
  - Ex: `lcd.Begin(16,2)`
- **print()** scrive una stringa su LCS
  - Ex: `lcd.print("hello, world!")`
- **setCursor()** imposta il cursore su un punto specifico dove scrivere
  - Ex: `lcd.setCursor(0,1)` imposta la colonna 0 e la riga 1
- **clear()** Questa funzione cancella qualsiasi testo o dato già visualizzato sul display LCD.
- **Home()** Imposta il cursore alla posizione iniziale (0,0)

# Circuito con LCD



Il display LCD necessita di sei pin di Arduino, tutti i pin devono essere output digitali.

Il display necessita anche di essere connesso ai 5V e alla messa a terra GND.

Il potenziometro viene utilizzato per controllare il contrasto del display.

# Hello World!

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 7, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

---

# Letture da seriale e scrittura su LCD

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 7, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  Serial.begin(9600);
}

void loop() {
  if(Serial.available()){
    delay(50);
    lcd.clear();

    while(Serial.available() > 0){
      lcd.write(Serial.read());
    }
  }
}
```