

# Comando di una Lampada Mediante Relè – (Tinkercad)

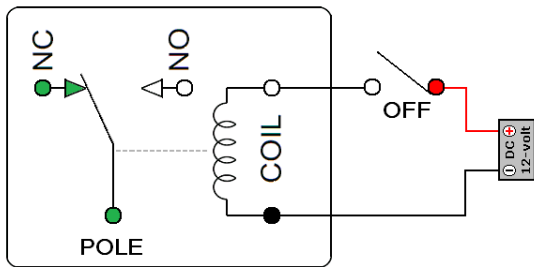
**Obiettivo:** Comando di una Lampada mediante un relè. Simulazione bastata sull'utilizzo del software Tinkercad.

## **Componenti elettronici:**

- Non servono componenti elettronici hardware, basta il tuo PC ed una connessione ad internet.
- <http://tinkercad.com/>

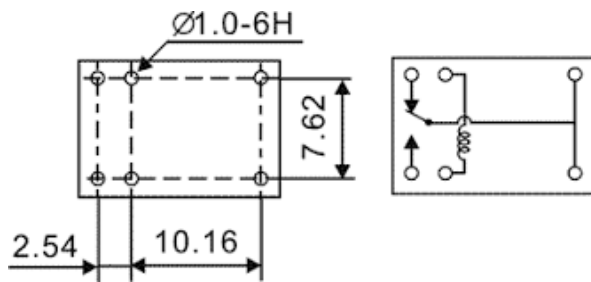
**Teoria:** Uno dei problemi principali di Arduino è legato alla impossibilità apparente di comandare dispositivi che richiedono tensione e/o correnti elevate. Infatti, è importante considerare che Arduino, attraverso le sue istruzioni di digitalWrite può generare su uno specifico pin in uscita una tensione massima di 5 Volt con una corrente pari a 70 milliAmpere. Molti utilizzatori si chiedono pertanto come sia possibile comandare dispositivi più complessi rispetto a quelli presenti nei tradizionali kit base Arduino, come lampade, ventole, o stripline led. La risposta a questa domanda è legata ad un semplice dispositivo elettromeccanico denominato relè (in inglese relay): il quale può essere utilizzato come un interruttore (ad alta tensione) comandato elettronicamente. Da un punto di vista fisico, il relè è costituito da un elettromagnete (costituito da una bobina di filo conduttore elettrico, tipicamente rame, avvolto intorno ad un nucleo di materiale ferromagnetico). Al passaggio di

corrente elettrica nella bobina, l'elettromagnete modificherà la posizione di un contatto mobile aprendo o chiudendo il circuito ad esso collegato. Il contatto aperto quando la bobina non è alimentata prende il nome di **normalmente aperto** (NO); mentre, l'altro contatto, quello chiuso prende il nome di **normalmente chiuso** (NC).



Esempio di funzionamento di un relèa

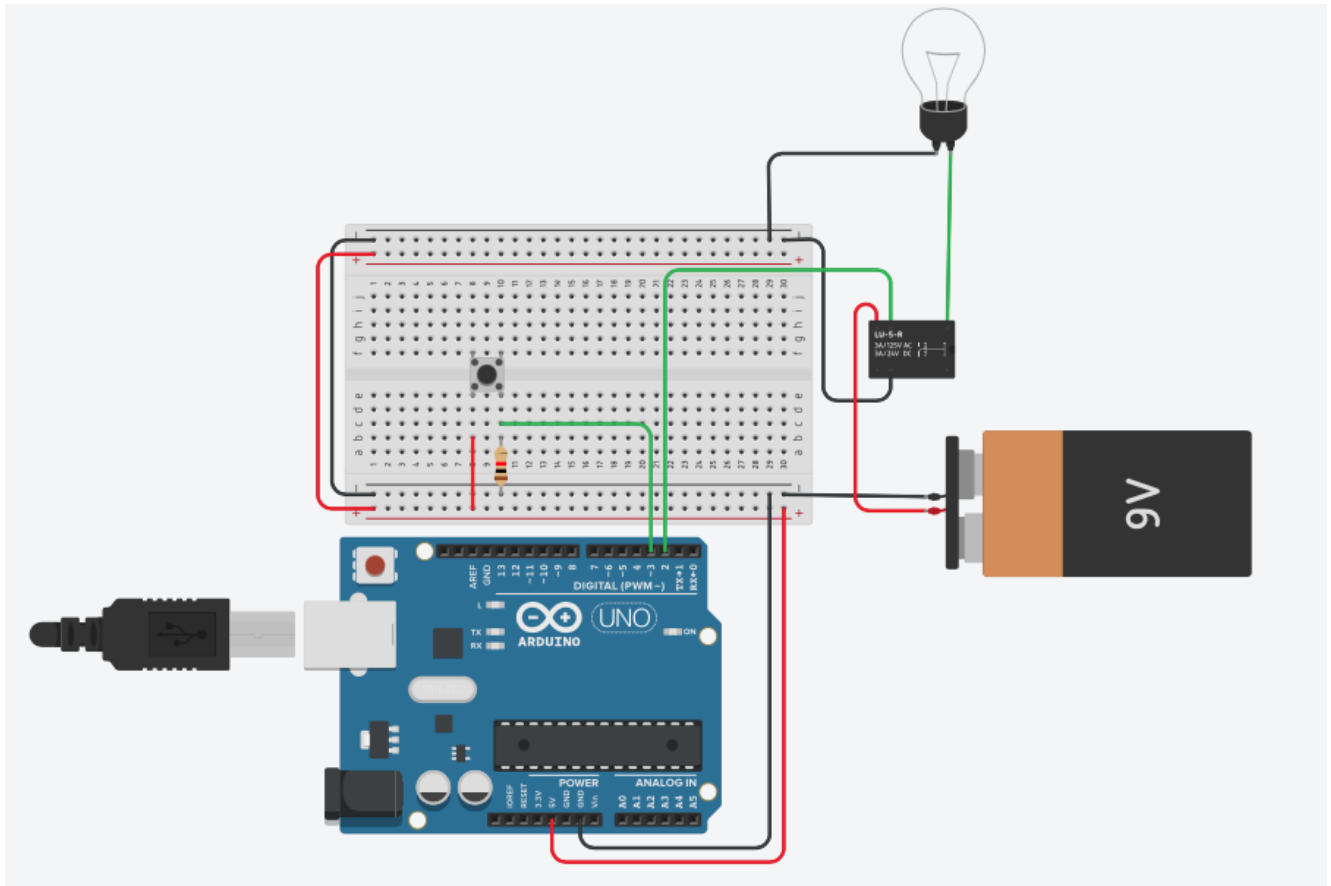
Da un punto di vista elettronico, utilizzare un relè per comandare una lampada a 230volts mediante Arduino è una procedura particolarmente semplice. Il primo passo è quello di collegare, mediante il contatto normalmente aperto del relè, la lampada ad una sorgente di alimentazione esterna (come ad esempio la rete elettrica di casa). In seguito la bobina del relè viene collegata al pin digitale di Arduino impiegato per il controllo della lampada. Questo ci permette di modificare la posizione del contatto mobile del relè attraverso l'istruzione di **digitalWrite** con la quale la bobina può essere o non essere eccitata. In questo modo è possibile comandare una lampada a 220Volts mediante una semplice tensione a 5Volts.



Datasheet del Relè LU-5-R presente in Tinkecad

Nel caso specifico, l'impiego di un pulsante permette di controllare l'attivazione della bobina e quindi il controllo della lampada. L'applicazione proposta è stata simulata mediante l'utilizzo del software di simulazione tinkercard. Tensioni elevate possono essere pericolose per la salute personale, pertanto l'impiego almeno in una fase iniziale di un simulatore rende questa operazione sicuramente più sicura. A seguire viene riportato lo schema elettrico ed il codice utilizzato per il comando della lampada mediante relè.

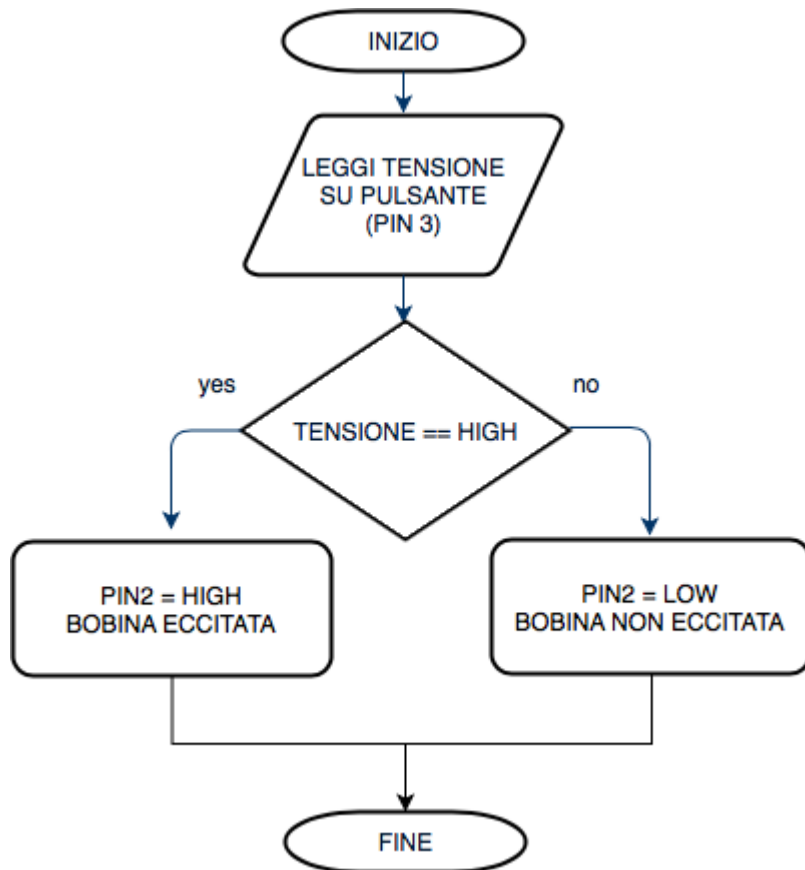
### **Collegamento Circuitale:**



Schema Circuitale

### Codice:

A seguire viene riportata la schematizzazione mediante flowchart dell'algoritmo utilizzato per realizzare il programma.



Flowchart

### Personalizzazioni:

E' possibile modificare il circuito utilizzando il pulsante come interruttore: la lampada deve rimanere accesa fino a quando il pulsante non viene premuto una seconda volta (Attenzione possibili problematiche di rimbalzo).

---

# Controllo di un LED mediante Monitor Seriale

**Obiettivo:** Utilizzare il monitor seriale per l'accensione/spegnimento di un LED

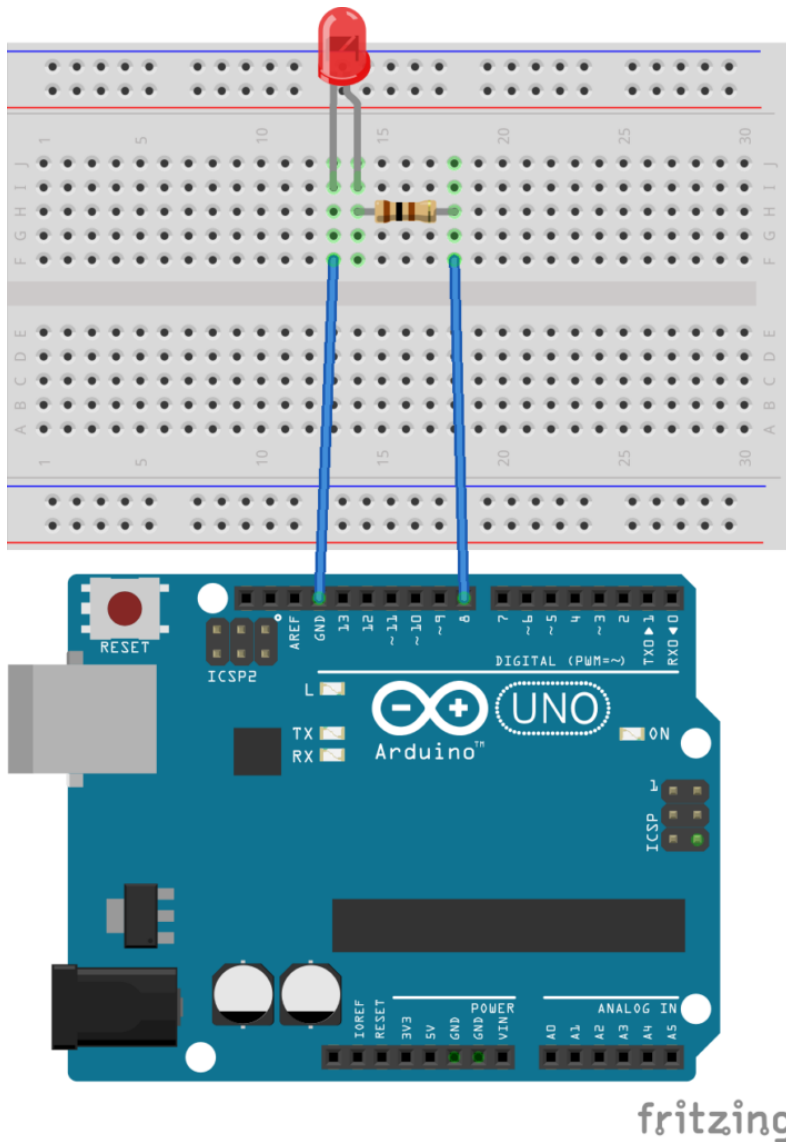
## **Componenti elettronici:**

- Arduino UNO
- Breadboard
- 1 Resistenza (100 Ohm) per LED
- 1 LED

**Teoria:** Il monitor seriale è uno degli strumenti più utilizzati per realizzare il “debug” delle applicazioni sviluppate sfruttando il microcontrollore Arduino. Attraverso il comando `Serial.println()` è infatti possibile stampare a monitor sia delle stringhe specifiche sia il contenuto delle variabili. Tuttavia, è importante considerare che attraverso il monitor seriale, non solo è possibile avviare una comunicazione **Arduino->PC**, ma anche una comunicazione **PC->Arduino**. Nel caso specifico di questa applicazione il monitor seriale sarà utilizzato per inviare al controllore Arduino un carattere (tipo char ad 8 bit) specifico per accendere o spegnere un LED.

**Collegamento Circuitale:** Viene in seguito riportato lo schema elettrico utilizzato per controllare un LED mediante Monitor Seriale. Il LED risulta semplicemente collegato al pin digitale 8 di Arduino, il quale sarà attivato o disattivato (HIGH/LOW) in funzione del carattere ricevuto dal monitor

seriale.



### Codice:

Le principali funzioni della classe **Serial** sono impiegate nel seguente codice:

- **Serial.available():**

Restituisce il numero di byte (caratteri) disponibili per la lettura nella porta seriale. Questi dati vengono

immagazzinati un buffer di ricezione seriale il quale può contenere 64 bytes.

- **Serial.read():**

Legge i dati provenienti dalla porta seriale

---

## **ARDWARE #16 Carica e Scarica di un Condensatore. Visualizzare i tempi con i LED.**

**Obiettivo** : Analizzare la carica e la scarica di un condensatore visualizzando i relativi tempi mediante due differenti LED

### **Componenti elettronici:**

- Arduino UNO
- Breadboard
- 2 led (un led verde per il tempo di carica ed uno rosso per il tempo di scarica)



- 1 Condensatore (1000 microFarad)
- 1 Resistenza (1000 Ohm) per carica/scarica condensatore
- 2 Pulsanti

**Teoria:** Il condensatore è un dispositivo elettronico che ha la capacità di immagazzinare carica elettrostatica. Tale dispositivo modifica il suo comportamento in funzione del segnale di alimentazione utilizzato nell'applicazione in questione (e.g., segnale continuo, segnale alternato, etc ...).

In regime di tensione costante (corrente continua), il condensatore si carica inizialmente (**regime transitorio**) fino a raggiunge una situazione di equilibrio dove la carica sulle armature corrisponde esattamente alla tensione applicata mediante il generatore (**regime stazionario**). Una volta carico, il condensatore si comporta come un circuito aperto ovvero interrompe ogni flusso di corrente all'interno del circuito. Questo poiché il legame tensione corrente in un condensatore è dettato dalla seguente relazione:

$$I(t) = C \cdot dV(t)/dt$$

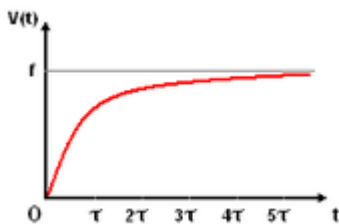
Pertanto essendo la tensione costante la derivata di tale valore sarà zero e di conseguenza anche l'intensità di corrente.

Al cessare dell'eccitazione sul circuito l'energia elettrica accumulata nel condensatore torna a scaricarsi sotto forma di corrente elettrica rilasciata nel circuito.

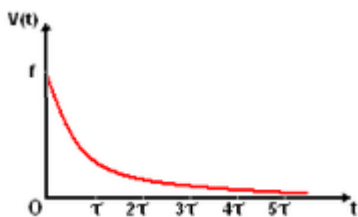
Il tempo di carica e di scarica dipende dalla costante di tempo  $\tau$  definita come:

$$\tau = R \cdot C$$

Nelle seguenti figure sono riportati i grafici relativi alla carica e alla scarica del condensatore. Nell'immagine di sinistra il condensatore, inizialmente scarico, si carica fino a raggiungere il livello di tensione finale  $f$ . Differentemente, nell'immagine di destra il condensatore, inizialmente carico e con una tensione  $f$  ai suoi capi, si scarica fino a raggiungere un livello di tensione finale pari a zero volt.



Carica di un condensatore

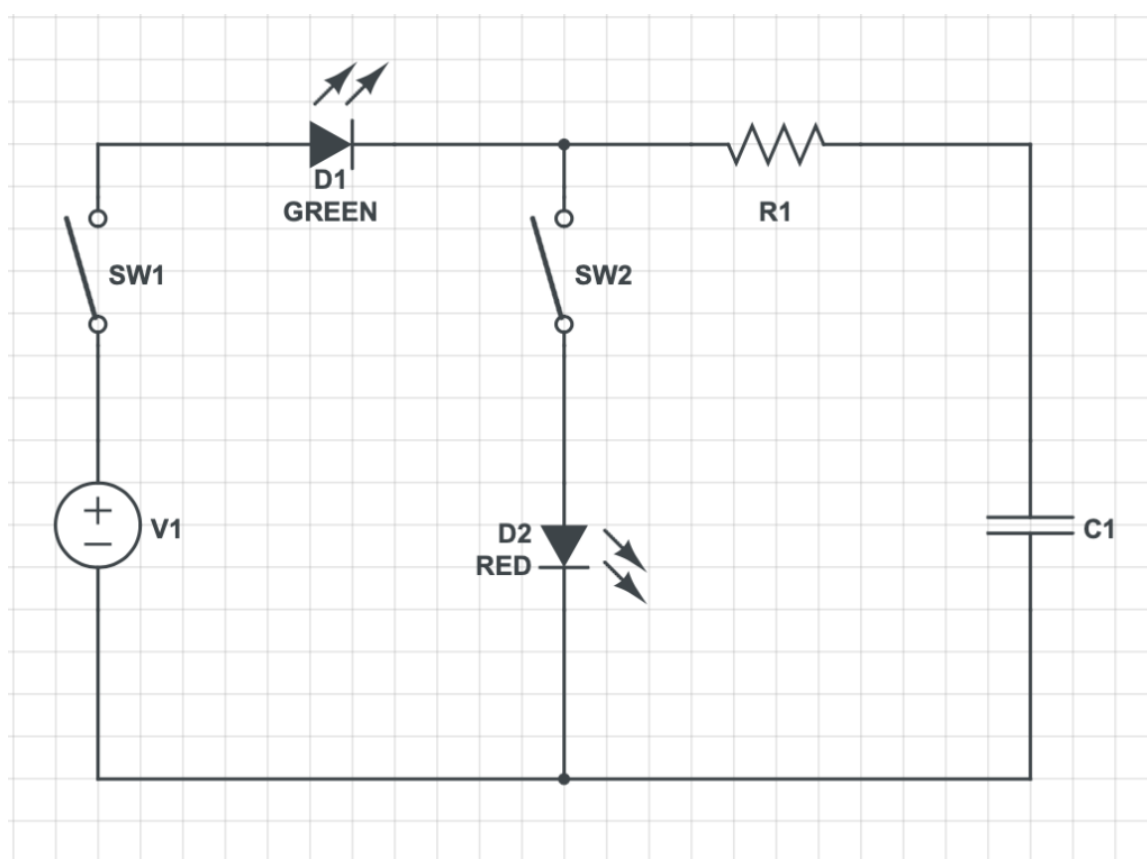


Scarica di un condensatore

In questa esperienza si propone l'utilizzo di due differenti led (un led verde ed uno rosso) per visualizzare i tempi di carica e di scarica del condensatore.

Il principale vantaggio di questo approccio è legato all'utilizzo del mediatore visivo al fine di comprendere il

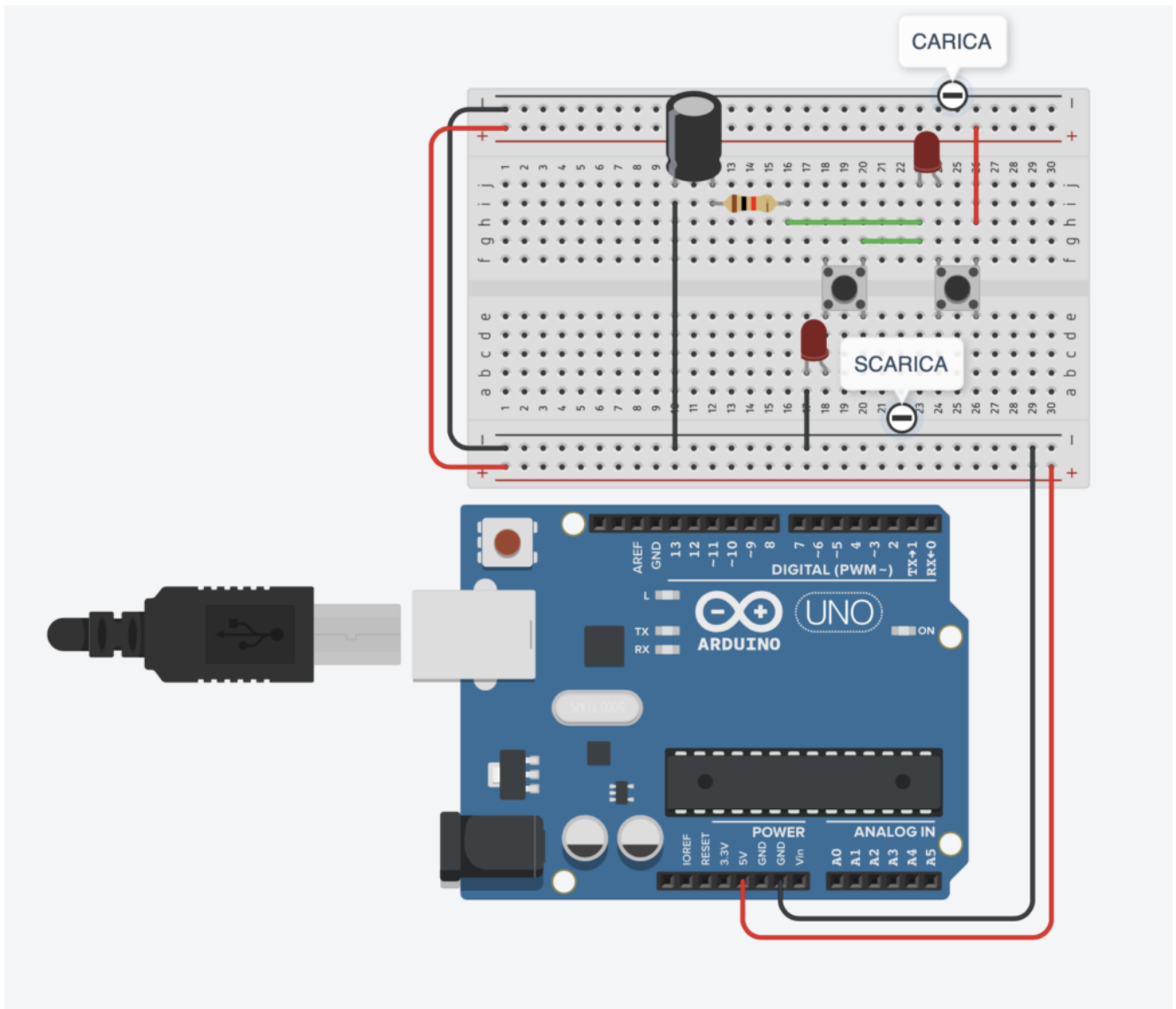
comportamento di un condensatore. Nello specifico, premendo un il pulsante SW1 il condensatore inizierà a caricarsi e pertanto il led D1 si accenderà poiché percorso da corrente. Una volta carico il condensatore non permetterà più il passaggio di corrente e pertanto il led D1 si spegnerà. A questo punto sarà possibile premere il pulsante SW2 al fine di scaricare il condensatore. In questo caso ad accendersi sarà il led D2 che rimarrà acceso fino a che il condensatore non si sarà scaricato.



Schema Circuitale

### Collegamento Circuitale:

Viene in seguito riportato lo schema elettrico utilizzato per analizzare il tempo di carica e di scarica di un condensatore.



Collegamento Circuitale

**Tinkercad:**

**Personalizzazioni:**

E' possibile modificare i tempi di carica e di scarica variando il valore della resistenza o del condensatore.

---

# Carica di un Condensatore

**Obiettivo** : Analizzare la carica di un condensatore mediante una lettura analogica.

## **Componenti elettronici:**

- Arduino UNO
- Breadboard
- 1 Condensatore (1000 microFarad)
- 1 Resistenza (100 Ohm) per carica condensatore
- 1 Pulsante
- 1 Resistenza (1 kOhm) per resistenza pulsante

**Teoria**: Il condensatore è un dispositivo elettronico che ha la capacità di immagazzinare carica elettrostatica. Tale dispositivo modifica il suo comportamento in funzione del segnale di alimentazione utilizzato nell'applicazione in questione (e.g., segnale continuo, segnale alternato, etc ...).

In regime di tensione costante (corrente continua), il condensatore si carica inizialmente (**regime transitorio**) fino a raggiunge una situazione di equilibrio dove la carica sulle armature corrisponde esattamente alla tensione applicata mediante il generatore (**regime stazionario**). Una volta carico, il condensatore si comporta come un circuito aperto ovvero interrompe ogni flusso di corrente all'interno del circuito. Al cessare dell'eccitazione sul circuito l'energia elettrica accumulata nel condensatore torna a scaricarsi sotto forma di

corrente elettrica rilasciata nel circuito.

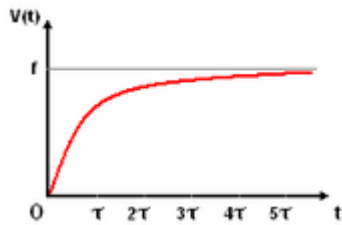
Il tempo di carica e di scarica dipende dalla costante di tempo *Tau* definita come:

$$\text{Tau} = R * C$$

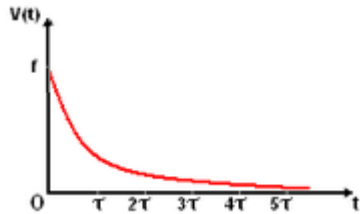
Nelle seguenti figure sono riportati i grafici relativi alla carica e alla scarica del condensatore. Nell'immagine di sinistra il condensatore, inizialmente scarico, si carica fino a raggiungere il livello di tensione finale *f*. Differentemente, nell'immagine di destra il condensatore, inizialmente carico e con una tensione *f* ai suoi capi, si scarica fino a raggiungere un livello di tensione finale pari a zero volt.

In questa esperienza si propone l'utilizzo del monitor seriale per visualizzare i tempi di carica del condensatore. E' importante considerare che i dati riportati sul monitor seriale (tempo e tensione) possono essere facilmente copiati in un foglio excel per ottenere una migliore rappresentazione grafica dell'evoluzione della tensione nel tempo.

L'utilizzo di un pulsante permette di avviare la carica del condensatore mediante l'utilizzo di una istruzione di digitalWrite.



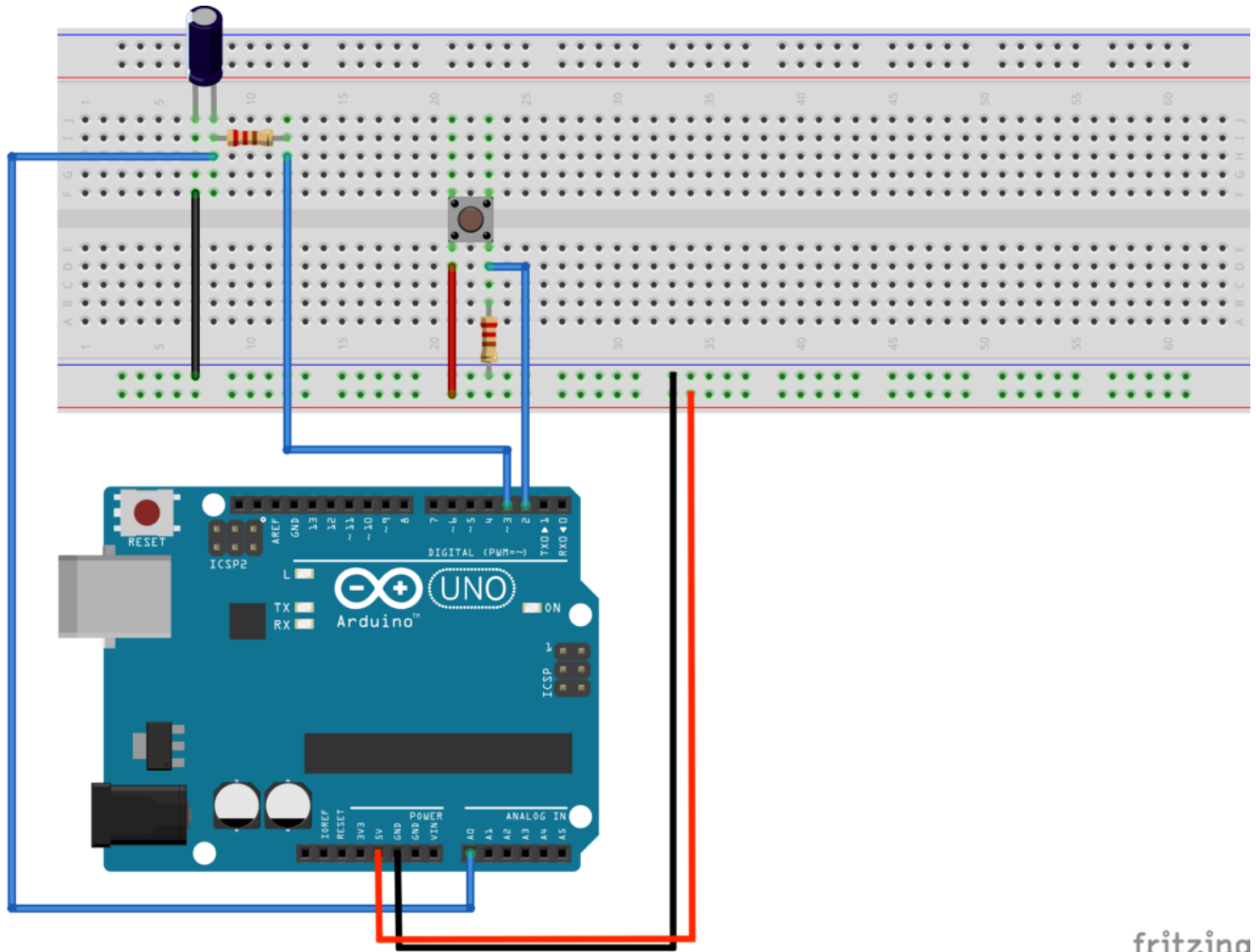
Carica di un condensatore



Scarica di un condensatore

### Collegamento Circuitale:

Viene in seguito riportato lo schema elettrico utilizzato per analizzare il tempo di carica di un condensatore. Nel dettaglio il tempo di carica viene monitorato utilizzando la lettura analogica del controllore Arduino.



fritzing

Collegamento Circuitale

**Codice:**

**Personalizzazioni:**

E' possibile modificare il codice e l'hardware per simulare anche la scarica del condensatore.



---

# Controllo di Temperatura e Umidità mediante DHT11

**Obiettivo:** Realizzare un controllo di temperatura e umidità mediante il sensore DHT11.

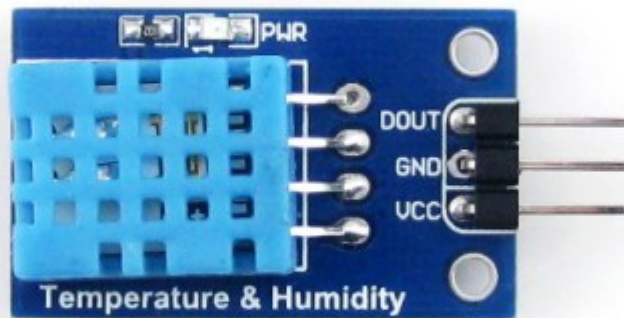
## **Componenti elettronici:**

- Arduino UNO
- DHT11

**Teoria:** Il DHT11 è un sensore digitale per la misura della temperatura ed umidità molto stabile e semplice da usare e configurare.

E' un sensore di basso costo che ha un range di misurazione dell'umidità che va da 20%RH al 90%RH (con una precisione di 5%RH) e di temperatura da 0 a 50°C (con una precisione di 2°C)

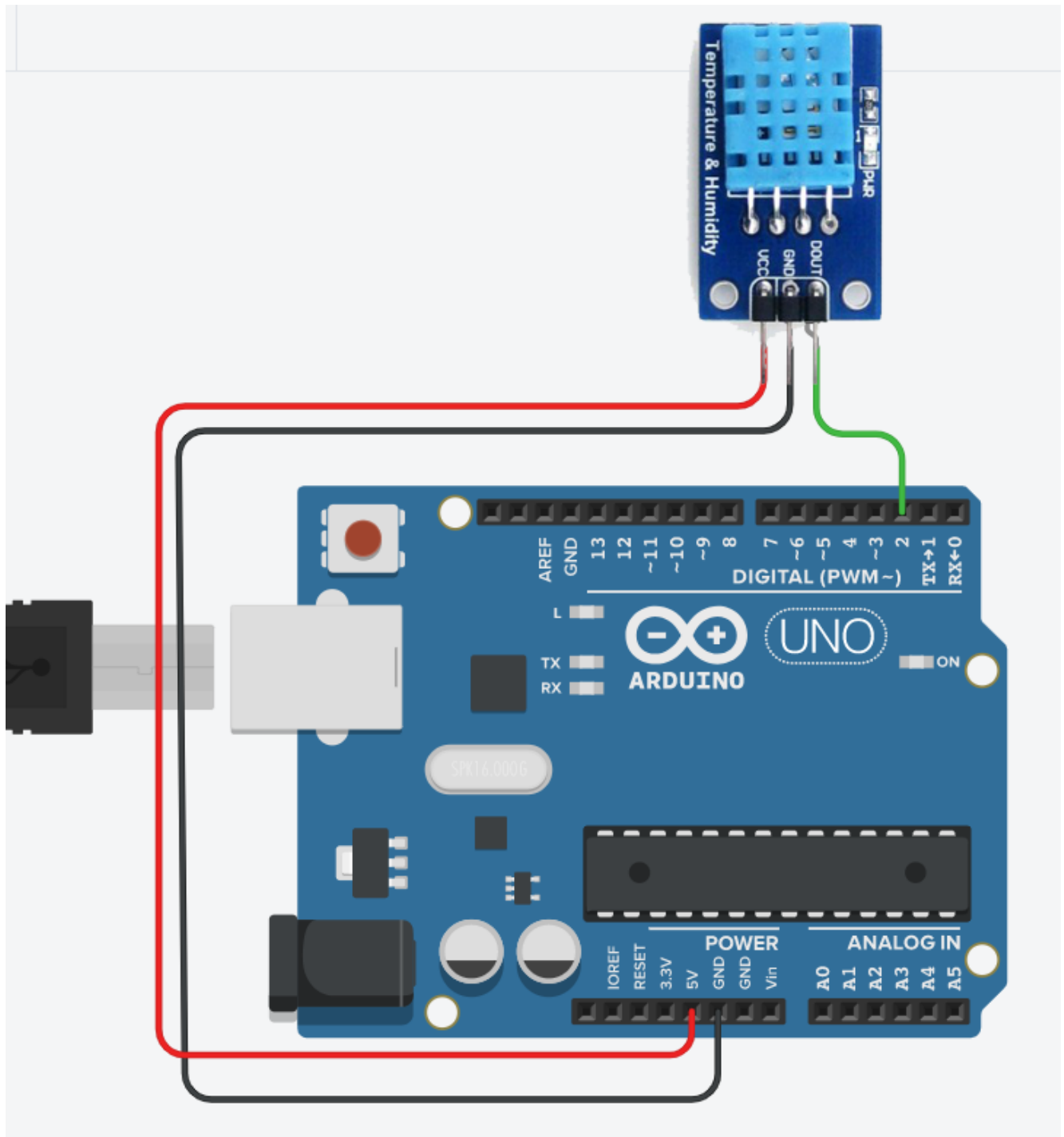
Il sensore si presenta con tre pin.



Guardando il dispositivo dal lato della parte sensibile, si ha che il pin di sinistra è quello dell'alimentazione (5V), quello centrale è il GND e il destro è il PIN del segnale digitale, da collegare ad un'uscita digitale di Arduino.

### **Collegamento Circuitale:**

Viene in seguito riportato lo schema elettrico utilizzato per valutare la temperatura mediante il dispositivo elettronico DHT11



### Codice:

Per il funzionamento del dispositivo è necessario caricare la libreria DHT.h nell'ide. Successivamente va definito il tipo di dispositivo che si sta utilizzando (nel nostro caso DHT11). Tale informazione va inserita nella dichiarazione dell'oggetto DHT ( `DHT dht(DHTPIN, DHTTYPE);` ).

La proprietà *readTemperature* restituisce la temperatura in gradi Celsius come default, a meno che non si inserisce un valore booleano *true* come argomento. In questo caso la temperatura verrà espressa in gradi Fahrenheit.

La proprietà *readHumidity* restituisce l'umidità relativa (RH).

### **Personalizzazioni:**

E' possibile modificare il circuito aggiungendo un display per visualizzare i valori di temperatura e umidità.

---

# **Misura di Temperatura mediante TMP36 [Tinkercad]**

**Obiettivo:** Realizzare un controllo di temperatura mediante il dispositivo TMP36. Il TMP36 è il sensore di temperatura presente sul simulatore tinkercad.

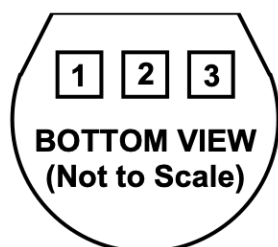
### **Componenti elettronici:**

- Arduino UNO
- Breadboard
- TMP36

**Teoria:** Il componente elettronico TMP36 è un dispositivo integrato ad alta precisione utilizzato per misurare la temperatura ambientale.

Dato il basso costo e l'ampia scala di valori ammissibili (ovvero da  $-40^{\circ}\text{C}$  fino a  $125^{\circ}\text{C}$ ) questi dispositivi sono particolarmente diffusi. E' inoltre importante considerare che non è necessaria nessuna operazione calibrazione per ottenere valori di accuratezza pari a  $\pm 1^{\circ}\text{C}$  ad una temperatura di circa  $+25^{\circ}\text{C}$  e  $\pm 2^{\circ}\text{C}$  nel range di temperature  $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ .

Questo dispositivo è caratterizzato da tre differenti pin ed un corpo semi-cilindrico. Guardando il lato piatto del dispositivo, il pin di sinistra è l'alimentazione (5V), il pin di destra la massa (GND), mentre sul pin centrale viene generata una tensione funzione della temperatura. La temperatura può essere pertanto misurata attraverso una lettura analogica sul pin centrale effettuata mediante il controllore Arduino.



PIN 1,  $+V_S$ ; PIN 2,  $V_{OUT}$ ; PIN 3, GND

00337-004

*Figure 4. T-3 (TO-92)*

TMP36 Package

Come riportato in precedenza è possibile utilizzare un pin di input analogico per ottenere il valore di temperatura mediante l'istruzione di `analogRead`. Nel caso specifico, osservando il grafico che riporta la caratteristica tensione/temperatura (per il **TMP36** la linea è evidenziata in rosso) per una tensione di uscita di 0.5V il sensore rileva la temperatura di 0°C. Pertanto valori di tensione inferiori a 0.5V indicano una temperatura sotto lo zero, mentre valori di tensione superiori a 0.5V indicano una temperatura positiva. Inoltre, è importante considerare che "una variazione di grado corrisponde ad una variazione di tensione di 10mV". Quindi, se sul pin di input analogico sono presenti 550mV significa che il sensore sta rilevando una temperatura di 5°C ( $550\text{mV} - 500\text{mV} = 50\text{ mV}$  variazione di 5°C).

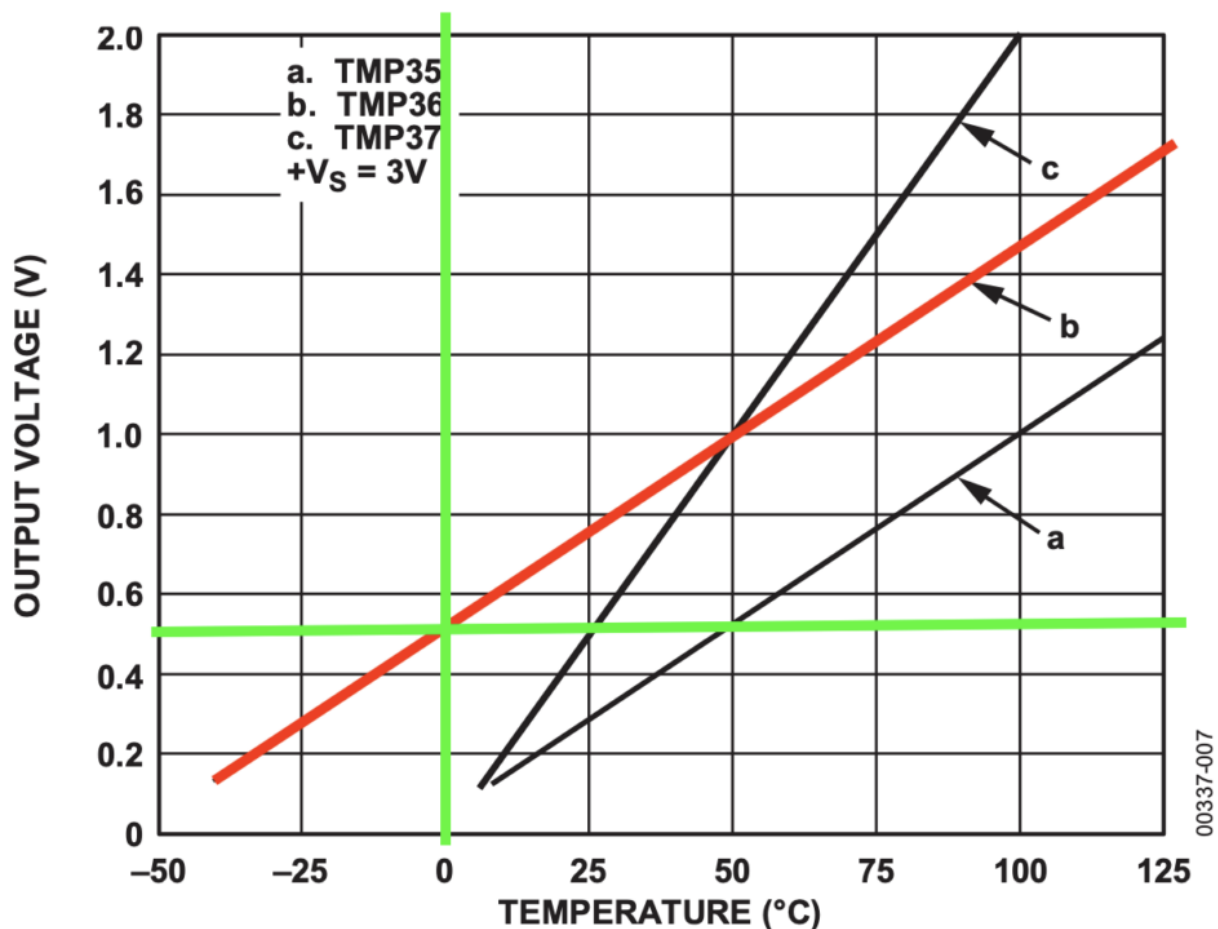


Figure 6. Output Voltage vs. Temperature

Viene in seguito riportato lo schema elettrico utilizzato per valutare la temperatura mediante il dispositivo elettronico TMP36

La temperatura viene determinata attraverso una lettura analogica ed una opportuna scalatura.

Nel dettaglio, la tensione prodotta dal componente TMP36 viene letta utilizzando il pin analogico A0 e mappata in un intervallo 0-1023. Considerando che alla temperatura di 0 gradi la tensione misurata è pari a 0.5V e che per ogni grado percepito si ha un incremento di tensione di 10mV è opportuno riportare il valore letto attraverso la funzione `analogRead` in mV ed eseguire un'opportuna scalatura. Pertanto se il valore letto mediante l'istruzione di `analogRead` è memorizzato in una variabile denominata `valTMP` la temperatura può essere ottenuta mediante la seguente formula:

$$temperatura = (((valTMP5.0)/1023.0)-0.5)*100$$

## **Tinkercad**

### **Personalizzazioni:**

E' possibile modificare il circuito aggiungendo una ventola che si accende in modo automatico superata una determinata temperatura.

---



# Controllo di Temperatura mediante LM35

**Obiettivo:** Realizzare un controllo di temperatura mediante il dispositivo LM35.

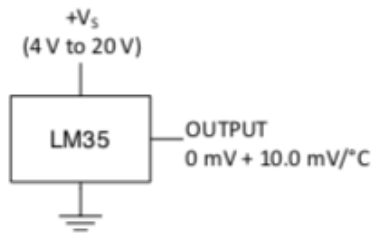
## **Componenti elettronici:**

- Arduino UNO
- Breadboard
- LM35

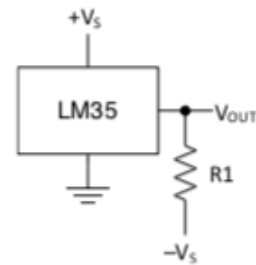
**Teoria:** Il componente elettronico LM35 è un dispositivo integrato ad alta precisione utilizzato per misurare la temperatura ambientale.

Dato il basso costo e l'ampia scala di valori ammissibili (ovvero da  $-55^{\circ}\text{C}$  fino a  $150^{\circ}\text{C}$ ) questi dispositivi sono particolarmente diffusi. E' inoltre importante considerare che nessun tipo di calibrazione esterna è richiesta.

### Basic Centigrade Temperature Sensor (2°C to 150°C)



### Full-Range Centigrade Temperature Sensor

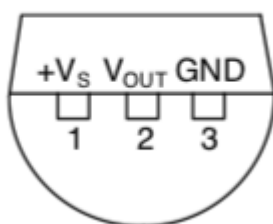


Choose  $R_1 = -V_S / 50 \mu\text{A}$   
 $V_{OUT} = 1500 \text{ mV}$  at  $150^\circ\text{C}$   
 $V_{OUT} = 250 \text{ mV}$  at  $25^\circ\text{C}$   
 $V_{OUT} = -550 \text{ mV}$  at  $-55^\circ\text{C}$

### Estratto Datasheet LM35

Questo dispositivo è caratterizzato da tre differenti pin ed un corpo semi-cilindrico. Guardando il lato piatto del dispositivo, il pin di sinistra è l'alimentazione (5V), il pin di destra la massa (GND), mentre sul pin centrale viene generata una tensione funzione della temperatura (10mV per ogni grado sopra lo zero). La temperatura può essere pertanto misurata attraverso una lettura analogica sul pin centrale effettuata mediante il controllore Arduino.

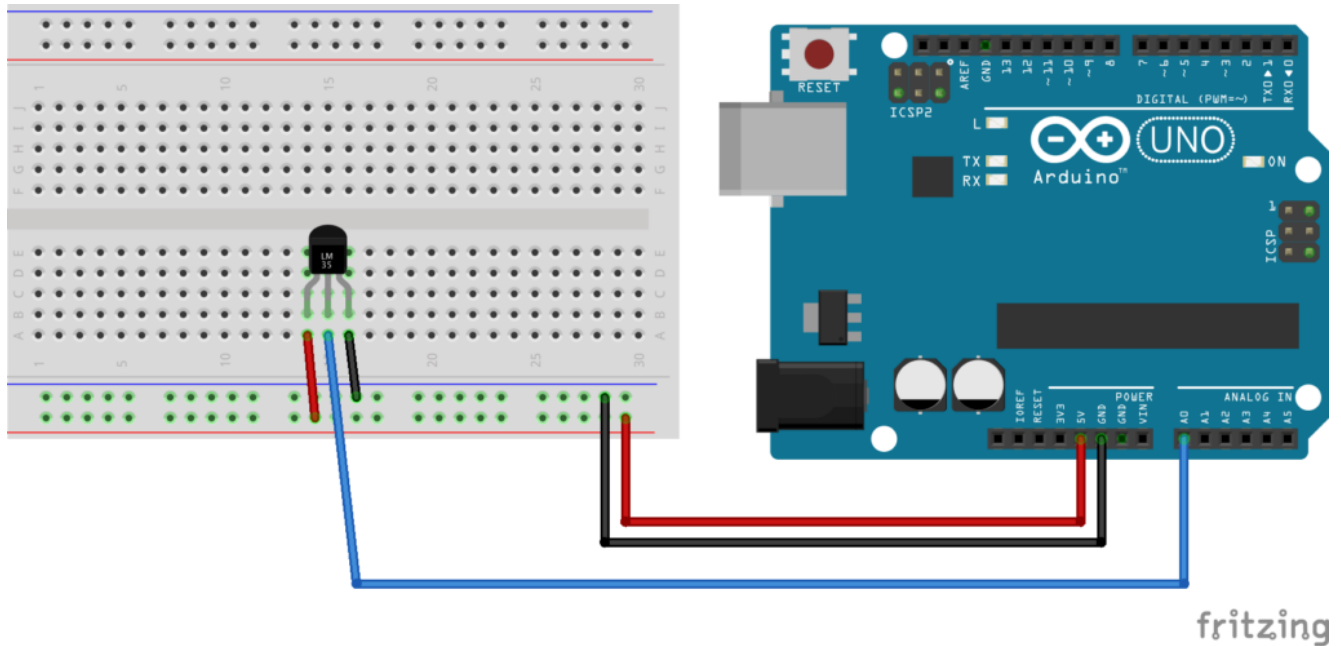
LP Package  
3-Pin TO-92  
(Bottom View)



### LM35 Package

### Collegamento Circuitale:

Viene in seguito riportato lo schema elettrico utilizzato per valutare la temperatura mediante il dispositivo elettronico LM35



## Collegamento Circuitale

### Codice:

La temperatura viene determinata attraverso una lettura analogica ed una opportuna scalatura.

Nel dettaglio, la tensione prodotta dal componente LM35 viene letta utilizzando il pin analogico A0 e mappata in un intervallo 0-1023. Considerando che per ogni grado percepito si ha un incremento di tensione di 10mV è opportuno riportare il valore letto attraverso la funzione `analogRead` in mV. Questa operazione può essere svolta dividendo il valore letto per 1023 e moltiplicando il risultato per 5000. I gradi sono infine ottenuti dividendo il risultato per 10.

### **Personalizzazioni:**

E' possibile modificare il circuito aggiungendo una ventola

che si accende in modo automatico superata una determinata temperatura.

---

# Controllare un LED mediante Smartphone

**Obiettivo:** Controllare un Led mediante Smartphone

**Pre-requisiti:**

[\*App Inventor – Bluetooth HC06 – LED\*](#)

**Componenti elettronici:**

- Arduino UNO
- Breadboard
- 1 modulo Bluetooth HC06
- 1 Smartphone Android
- 1 Resistenza da 220 Ohm
- 1 LED

**Teoria:** Partendo dall'applicativo Android, sviluppato nella

lezione: “Appinventor – Bluetooth HC06 – LED” in questa lezione viene illustrato come utilizzare l’app creata per controllare un LED via smartphone attraverso il protocollo bluetooth.

Nel caso specifico il protocollo Bluetooth (abbreviato in BT) è uno standard di trasmissione dati per reti senza fili. Il BT fornisce un metodo standard, economico e sicuro per scambiare informazioni tra dispositivi diversi attraverso onde radio a corto raggio (qualche decina di metri). La comunicazione Bluetooth tra lo Smartphone ed Arduino avviene attraverso l'utilizzo di un modulo BT per Arduino denominato HC06.



## Modulo BT HC06

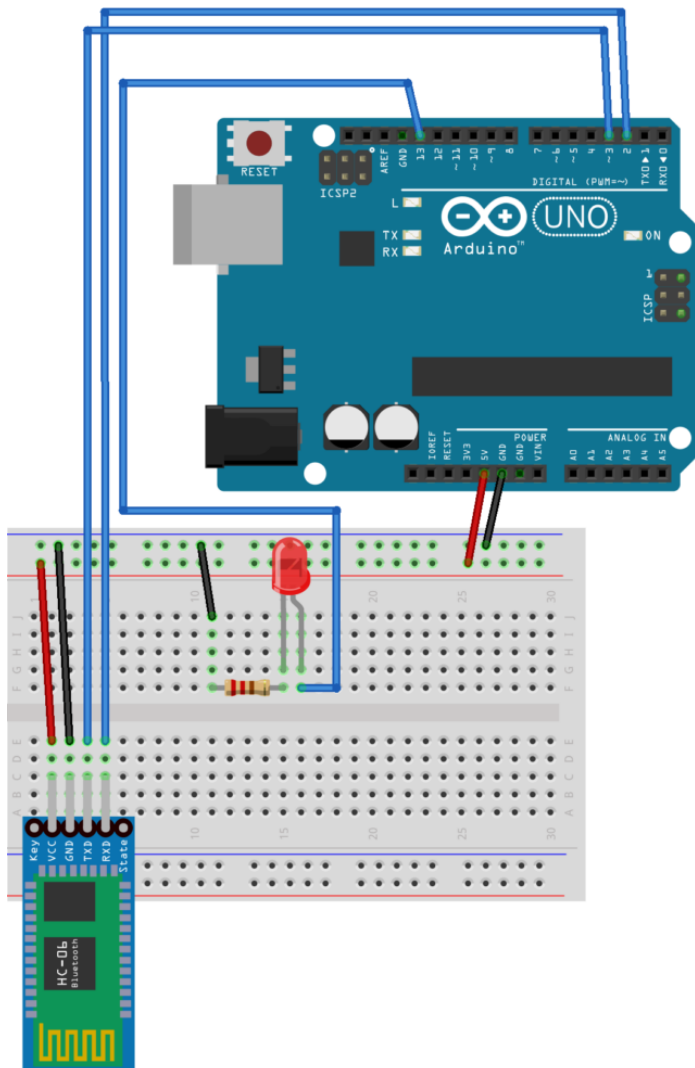
Il Modulo HC06 implementa un convertitore da porta seriale UART a porta Bluetooth permettendo la comunicazione tra un microprocessore come una scheda Arduino e un dispositivo dotato di comunicazione Bluetooth (PC, Smartphone o Tablet). Il modulo è dotato di quattro differenti PIN descritti nella seguente tabella:

PIN	Descrizione
1	Vcc (Alimentazione, funziona correttamente anche con i 5 V di Arduino)
2	Gnd (Ground)
3	Txd (Pin di trasmissione seriale. Deve essere collegato ad un pin di ricezione)
4	Rxd (Pin di ricezione seriale. Deve essere collegato ad un pin di trasmissione)

Da un punto di vista pratico, grazie all'impiego della specifica **libreria [SoftwareSerial](#)** inclusa nel pacchetto software di Arduino è possibile comunicare con il modulo Bluetooth ed il relativo dispositivo connesso. La libreria mette infatti a disposizione una classe **SoftwareSerial** all'interno della quale sono definite le principali funzioni necessarie per utilizzare il modulo HC06. Quali:

- **Begin:** Inizializza l'interfaccia seriale definendo la velocità standard di comunicazione
- **Available:** Indica se dei dati sono stati inviati al modulo HC06 (dati disponibili da leggere).
- **Read:** Legge i dati ricevuti
- **Write:** Scrive i dati

### Collegamento Circuitale:



fritzing

Schema Circuitale

**Codice:**

**Personalizzazioni:** E' possibile aggiungere più led modificando l'applicazione ed il codice sorgente.

---

# Il Motore passo-passo (Stepper)

**Obiettivo:** Pilotare un motore passo-passo tramite Arduino

## **Componenti elettronici:**

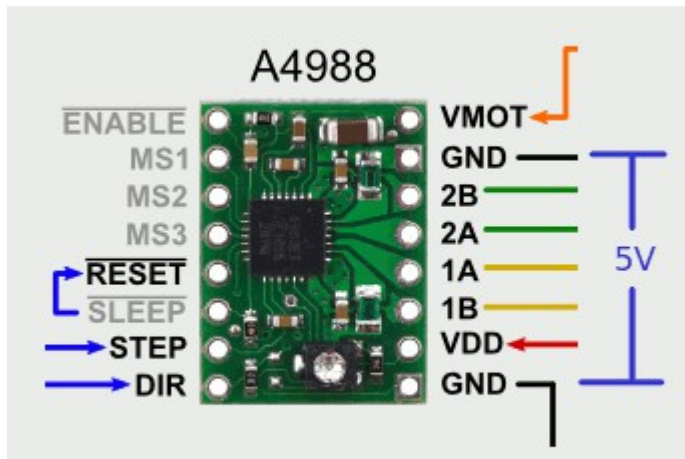
- Arduino UNO
- Breadboard
- 1 Motore Stepper (e.g. Nema17 1.7A 1.8° 42MM Stepper Motor – 42shdc3025-24b)
- 1 Driver A4988
- 1 Alimentatore a 12 V e 2 A a corrente continua

**Teoria:** Il motore passo-passo, detto anche stepper, è un motore elettrico sincrono in corrente continua, senza spazzole, che permette la suddivisione della rotazione in piccoli angoli detti step. E' un motore molto preciso e veloce e facilmente controllabile tramite una scheda elettronica, denominata "driver". Esso viene collegato ad Arduino, e rende il controllo del motore molto facile, permettendo, con soli due uscite digitali, di controllare la velocità, la direzione e l'angolo di rotazione.

Il driver è il vero protagonista del controllo del motore



passo-passo. Si pone tra Arduino e il motore e viene collegato come dal seguente schema:



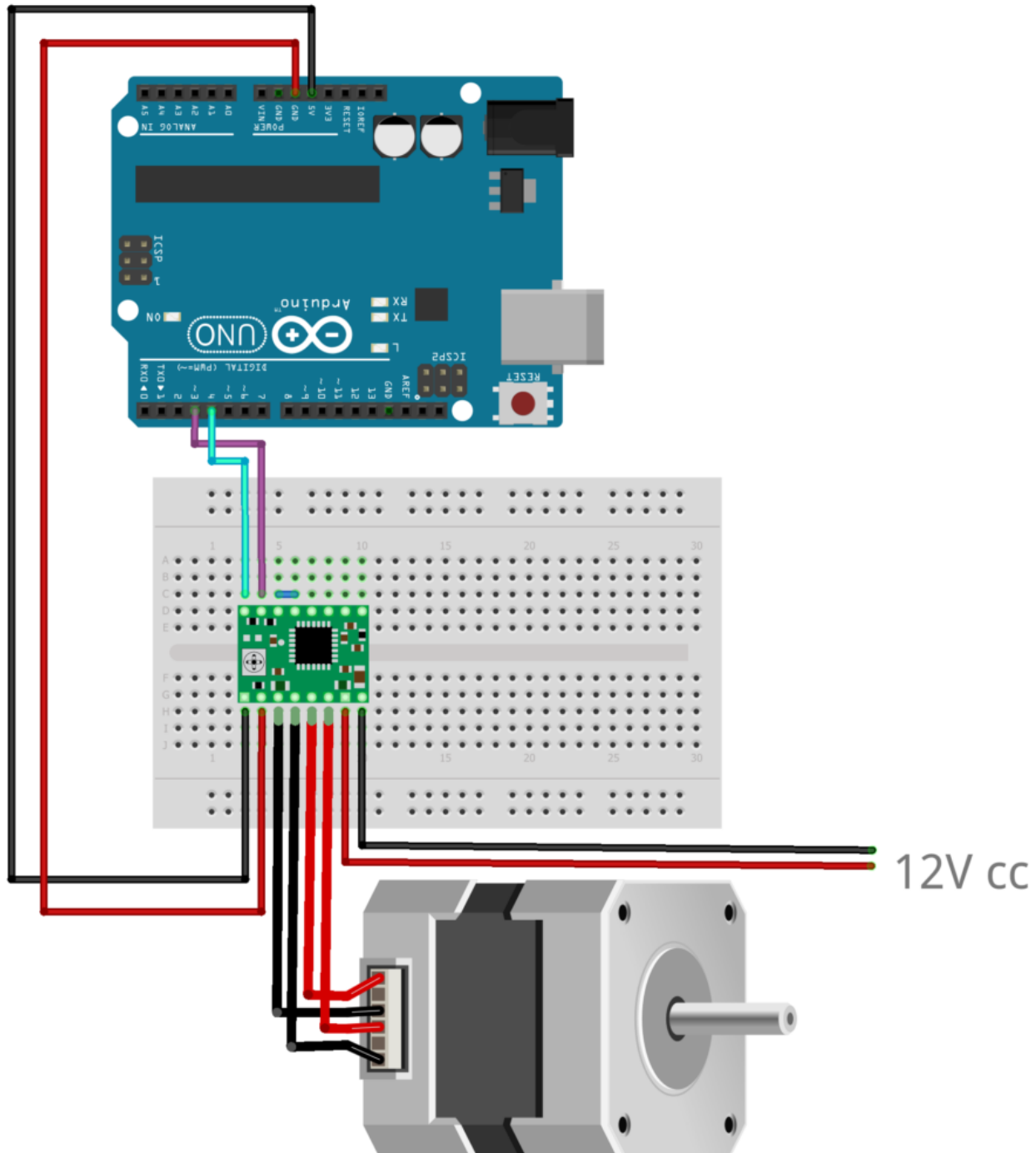
I pin STEP e DIR servono per controllare la rotazione (e la velocità) e la direzione e vanno collegati a due uscite digitali di Arduino. Sulla destra invece l'A4988 presenta i pin per il collegamento al motore. GND (in basso) e VDD servono per alimentare la scheda tramite Arduino. Fondamentali sono i pin 1A, 1B, 2A e 2B che vanno collegati alle fasi del motore, come descritto più avanti. Infine, i pin GND e VMOT riguardano l'alimentazione del motore e vanno collegati all'alimentatore a 12 V.

[4988-DatasheetScarica](#)

Il motore necessita di 200 impulsi per completare un giro. Il programma si basa su cicli che gestiscono la rotazione. La velocità di rotazione dipende dal tempo di attesa tra un impulso e un altro all'interno dei singoli cicli di rotazione.

La pausa tra un impulso e un altro (e quindi la gestione della velocità di rotazione) dipende dalla funzione *delayMicroseconds* presente all'interno del ciclo di rotazione.

### Collegamento Circuitale:



**Codice:**

**Personalizzazioni:** E' possibile collegare un potenziometro e un pulsante per gestire la direzione e la velocità di rotazione.

---

# Controllare un Servo Motore tramite Joystick

**Obiettivo:** Come controllare due servomotori utilizzando un Joystick per Arduino.

**Pre-requisiti**

[Il Servomotore](#)

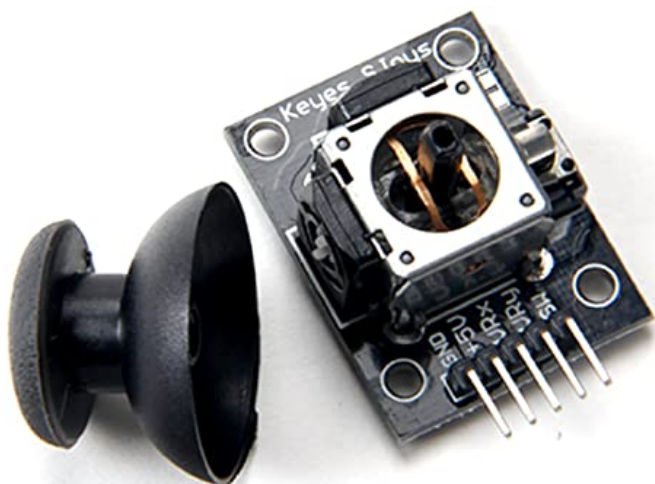
## Componenti elettronici:

- Arduino UNO
- Breadboard
- 2 Servomotori
- 1 Joystick KY-023K

**Teoria:** Il **Joystick** è un dispositivo elettronico che trasforma i movimenti di una leva manovrata dall'utente in una serie di segnali elettrici, questi segnali possono essere in seguito utilizzati per controllare un software (e.g., videogame), un'apparecchiatura o un attuatore meccanico. Possono esistere due differenti tipologie di Joystick:

- Joystick Digitale: Rileva solamente la direzione dell'inclinazione della leva.
- Joystick Analogico: Rileva anche l'ampiezza dell'inclinazione.

Nello specifico in questo articolo viene illustrato il funzionamento di uno dei controller più tipicamente utilizzati e presenti nei vari kit Arduino: il "Dual Axis Joystick Module **KY-023**", Questo dispositivo, basato sul controller della PlayStation2, utilizza due potenziometri bi-assiali per controllare l'asse X e l'asse Y. Inoltre è possibile premere il controller per attivare uno switch. Nello specifico, la tensione di funzionamento del dispositivo è compresa nel range 3.3 – 5 V. Mentre le dimensioni sono pari a 2.6 x 3.4 cm.



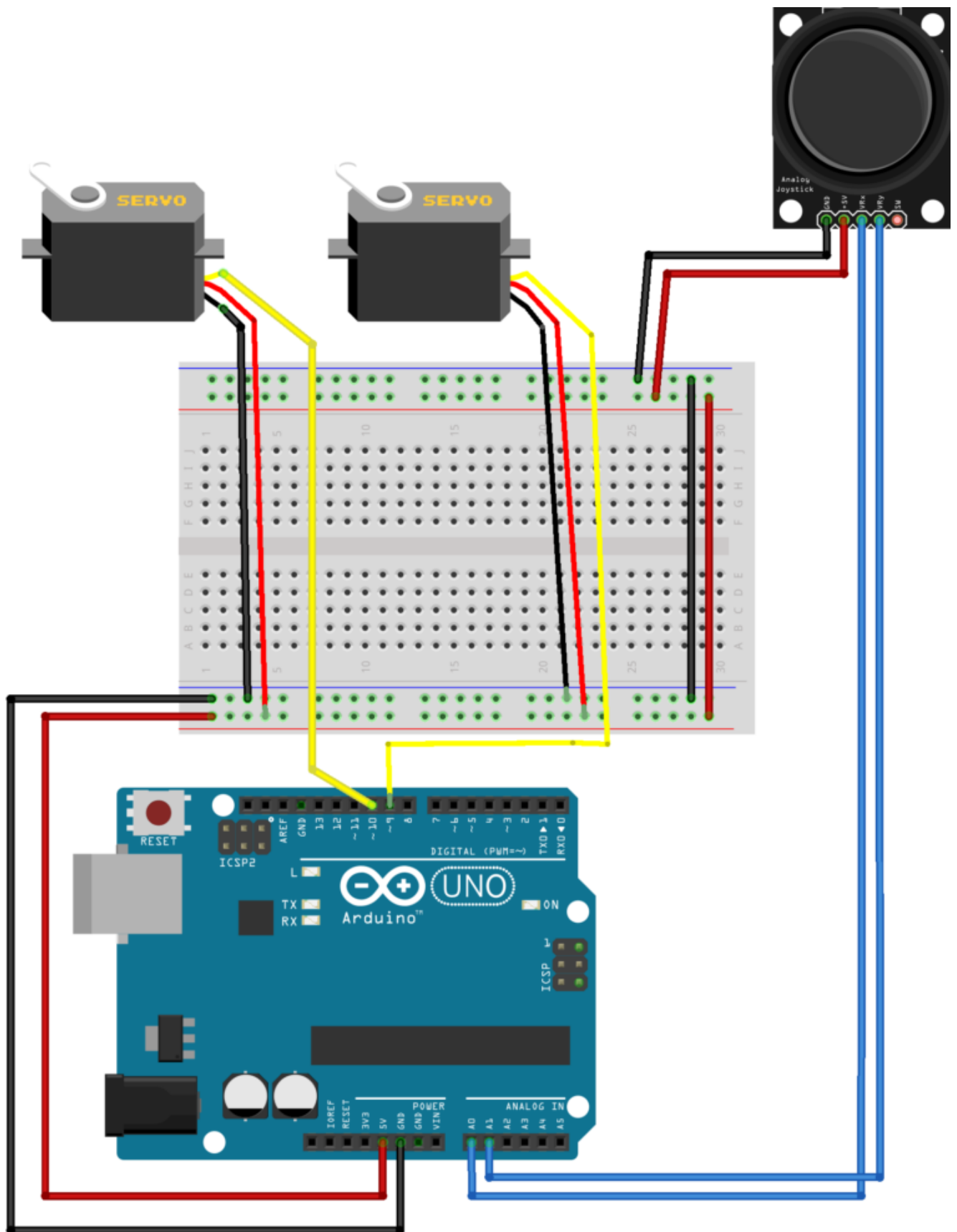
## Dual Axis Joystick Module **KY-023**

Nella tabella a seguire è riportata la piedinatura utile per collegare in modo corretto il Joystick KY-023.

<b>KY-023 PIN</b>	<b>Descrizione</b>
GND	Ground
+5V	Alimentazione Vcc
VRx	Uscita Analogica (Asse X)
VRy	Uscita Analogica (Asse Y)
SW	Switch

In questo articolo il Joystick viene utilizzato per comandare due differenti servomotori. Un servo è associato all'asse X ed un altro è invece associato all'asse Y. La posizione di riposo dei due servomotori è per entrambi 90 gradi. Spostando il Joystick lungo l'asse X si può modificare la posizione del servo associato all'asse X di un angolo variabile da 0 a 180 gradi. Lo stesso accade modificando la posizione del joystick lungo l'asse y.

## Collegamento Circuitale:



**Codice:**