

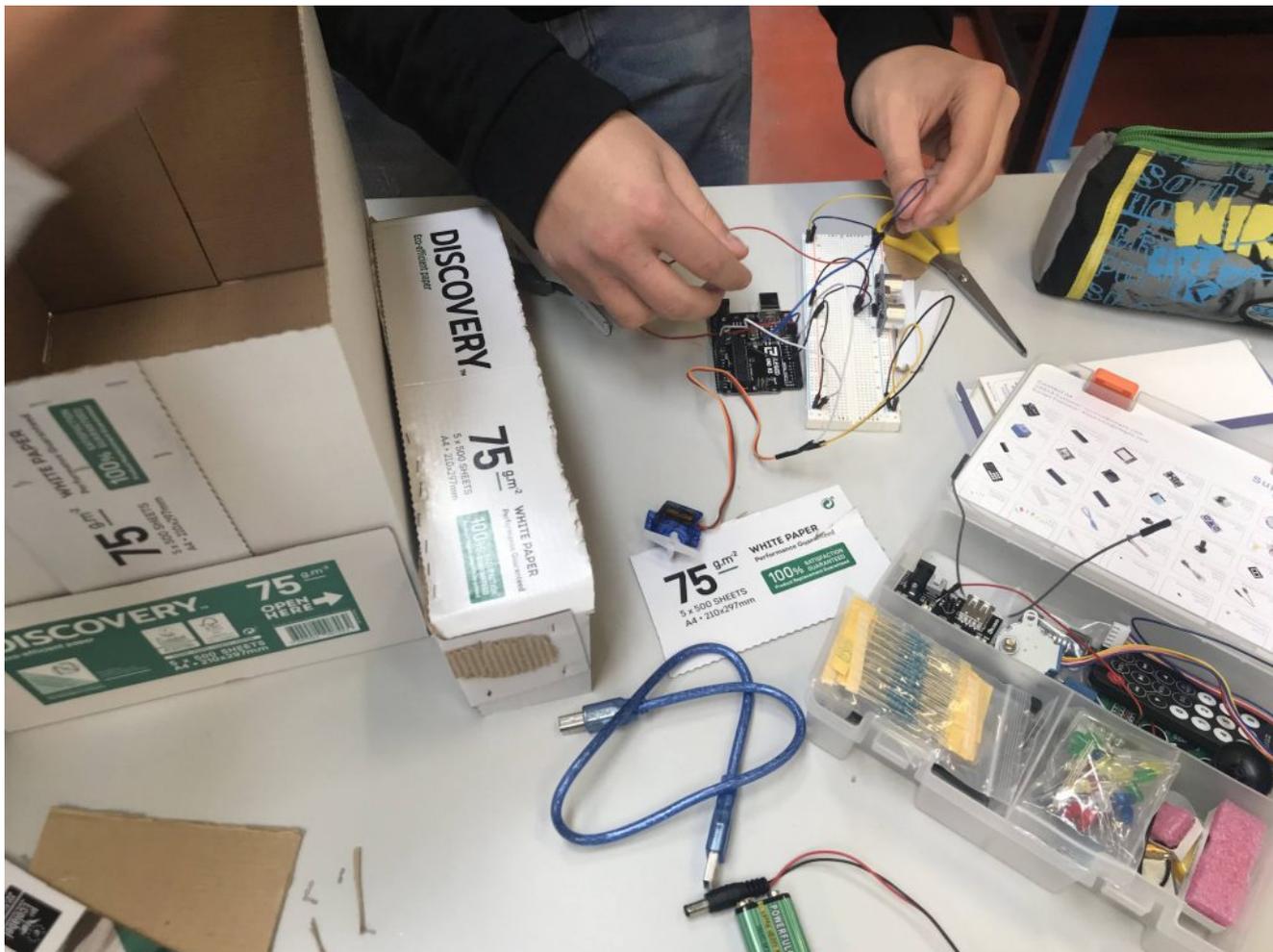
# Il Cestino Smart scomposto

L'ultimo decennio è stato contrassegnato da un aumento delle problematiche legate al rispetto dell'ambiente ed al risparmio energetico. Il riscaldamento del pianeta, la guerra in Ucraina e la conseguente crisi energetica, impongono tutti a ripensare alle abitudini quotidiane.

La scuola ha il dovere di intervenire sulle nuove generazioni, fornendo loro le competenze e la sensibilità necessaria per rispettare il nostro pianeta.

Noi di Arduino Facile siamo convinti che i piccoli gesti, quelli concreti, alla portata di tutti, siano quelli che maggiormente ci avvicinano alla realtà, che, a volte, ci sembra lontana da noi.

All'ITIS "E.Mattei" di Urbino (<https://www.itisurbino.edu.it/>), gli studenti della 3BMC, dell'indirizzo di Meccanica e Meccatronica, si sono cimentati nella realizzazione di un cestino "intelligente", che, alle pratiche igieniche, unisca ad esempio la raccolta differenziata dei rifiuti.



I gruppi al lavoro

Il punto di partenza è stato il nostro articolo <https://www.arduino facile.it/2021/05/13/il-cestino-smart/>

Successivamente la classe è stata divisa in gruppi di 4/5 studenti che hanno lavorato in modalità jigsaw: ogni componente si è specializzato in una particolare competenza, che poi ha condiviso con i compagni.

*Il resto è stato un mix di fantasia ed immaginazione...*

Ogni gruppo, infatti, ha interpretato a suo modo la consegna, ed ha elaborato una propria versione del *cestino smart*. Ne è nata una competizione agguerrita in cui ha avuto la meglio il

gruppo denominato "Marrakesh" che ha realizzato un cestino con apertura a due ante, e una spia ad indicare il riempimento.

Le istruzioni di montaggio sono disponibili nella seguente presentazione preparata dai componenti del gruppo

[cestino smartScarica](#)

Al secondo posto si è piazzata un'altra versione, ancora più amica dell'ambiente: un cestino formato maxi, con due ante, per differenziare i rifiuti

Infine, sul podio un cestino più classico con la segnalazione di apertura.

---

## **Antifurto con funzioni domotiche**

**Obiettivo:** Realizzare un antifurto con funzioni domotiche, con annesso un secondo Arduino in collegamento seriale in grado di comandare un cancello scorrevole.

Progetto realizzato dallo studente **Sanchini Simone**, dell'Università Politecnica delle Marche

## Componenti elettronici:

- 2 Arduino UNO
- 2 Breadboard
- 4 Resistenze da 220 Ohm per i led
- 4 LED
- 6 Interruttori
- 1 Resistenza (2.2kOhm) per Fotoresistenza
- 1 Fotoresistenza
- 2 Motori DC
- 1 Piezo
- 1 Sensore PIR
- 1 Sensore IR
- 1 Telecomando IR
- 1 KeyPad 4x4

## **Pre-requisiti:**

1. Password di accesso con Keypad 4x4 e Arduino – (Tinkercad)
2. Crepuscolare (Smart Lamp)
3. Controllo di un LED Mediante Telecomando ELEG00
4. Blinking Led Senza Delay: MILLIS()
5. Collegamento Seriale Arduino (non trovato su Arduinofacile)
6. Il Sensore di Presenza

## **Teoria:**

Come abbiamo detto, lo scopo dell'esercitazione è quello realizzare un antifurto con funzioni domotiche utilizzando poi un secondo Arduino in collegamento seriale in grado di comandare un cancello scorrevole.

### **ARDUINO 1:**

Attraverso il Keypad sarà possibile inserire tre tipi di codici diversi:

1. 1234: permette di inserire l'allarme (Stato=1), al suo inserimento le finestre e la porta di casa devono essere chiuse, queste ultime vengono simulate da due interruttori. All'inserimento ci saranno 5 secondi di delay in modo di permettere all'utente di uscire e chiudere la porta, dopodiché le luci presenti (simulate dal led) vengono spente, le tapparelle abbassate e attraverso il collegamento seriale viene dato il comando al secondo Arduino di aprire il cancello. Lo stop delle tapparelle e del cancello avviene attraverso dei finecorsa (simulati da due interruttori).
2. 1235: permette di togliere l'allarme (Stato=0), al momento del disinnescio, verranno aperte le tapparelle se giorno, oppure accese le luci se notte, utilizzando come riferimento il sensore crepuscolare.
3. 1236: permette di inserire l'allarme notturno (Stato=2), con la differenza dal primo che in questo caso esso non tiene in considerazione dell'apertura delle finestre. Al momento dell'inserimento attraverso il collegamento seriale viene dato il comando al secondo Arduino di chiudere il cancello.

Indicazioni LED:

Rosso: allarme inserito;

Rosso-Verde: allarme notturno inserito;

Rosso-Giallo-Verde (lampeggio): codice errato;

Giallo: finestre/porta aperta;

Cosa succede in caso di apertura di finestre o porta quando il nostro allarme è inserito?

Il sistema passerà allo stato di intrusione (Stato=3), settando così un timer realizzato dalla funzione millis() di tempo t, nella quale è possibile inserire il codice di sblocco; In caso il codice di sblocco non è inserito entro il tempo limite la "sirena" inizia a suonare finché l'antifurto non verrà sbloccato.

La stessa cosa succede in caso di intrusione dalle finestre quando l'allarme notturno è inserito.

ARDUINO 2:

Il secondo Arduino permette di comandare un cancello attraverso o i comandi mandati dal primo, oppure attraverso un telecomando IR, con un solo pulsante (a causa di un problema di lettura di tinkercad).

Premendo il pulsante lo stato cambierà lo stato del cancello in base a quello precedente:

Stato=0 -> Cannello fermo;

Stato=1 -> Cannello in chiusura;

Stato=2 -> Cannello fermo;

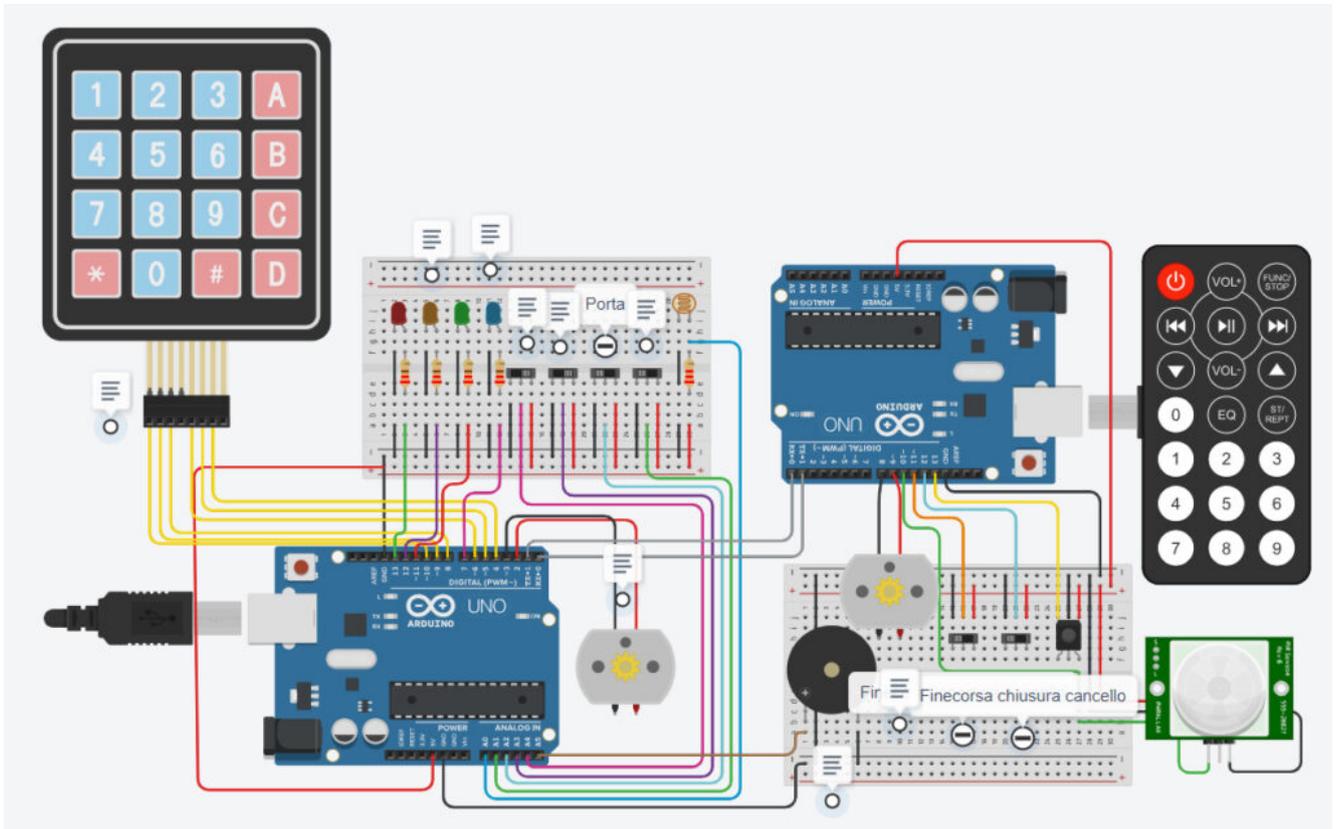
Stato=3 -> Cannello in apertura;

Il cancello è fermato in apertura e chiusura da due finecorsa, simulati da due interruttori.

Se in fase di chiusura il sensore PIR rileva qualcosa il cancello torna allo stato di apertura.

Quando il cancello finisce l'apertura, premendo il finecorsa, parte un timer di tempo  $t$ , ed alla fine di esso il cancello torna in fase di chiusura automatica.

**Collegamento Circuitale:**



**TINKERCAD:**

[https://www.tinkercad.com/things/3aMc1kLCsu0-definitivo-sanchini/editel?sharecode=\\_gcKkLKcUpJEY1lHHIETPLS8ldhHe4IcM9bkXZUF48k](https://www.tinkercad.com/things/3aMc1kLCsu0-definitivo-sanchini/editel?sharecode=_gcKkLKcUpJEY1lHHIETPLS8ldhHe4IcM9bkXZUF48k)

**Codice:**

---

# Le frecce dell'AUDI con Arduino

**Obiettivo:** Realizzare un sistema di controllo dei led che simula l'effetto delle frecce di un Audi. Le luci si accendono alla pressione di un pulsante.

## **Componenti elettronici:**

- Arduino UNO
- Breadboard
- 8 Resistenze da 220 Ohm per i led
- 4 LED
- 2 Resistenza da 10K0hm per i pulsanti
- 2 Pulsanti

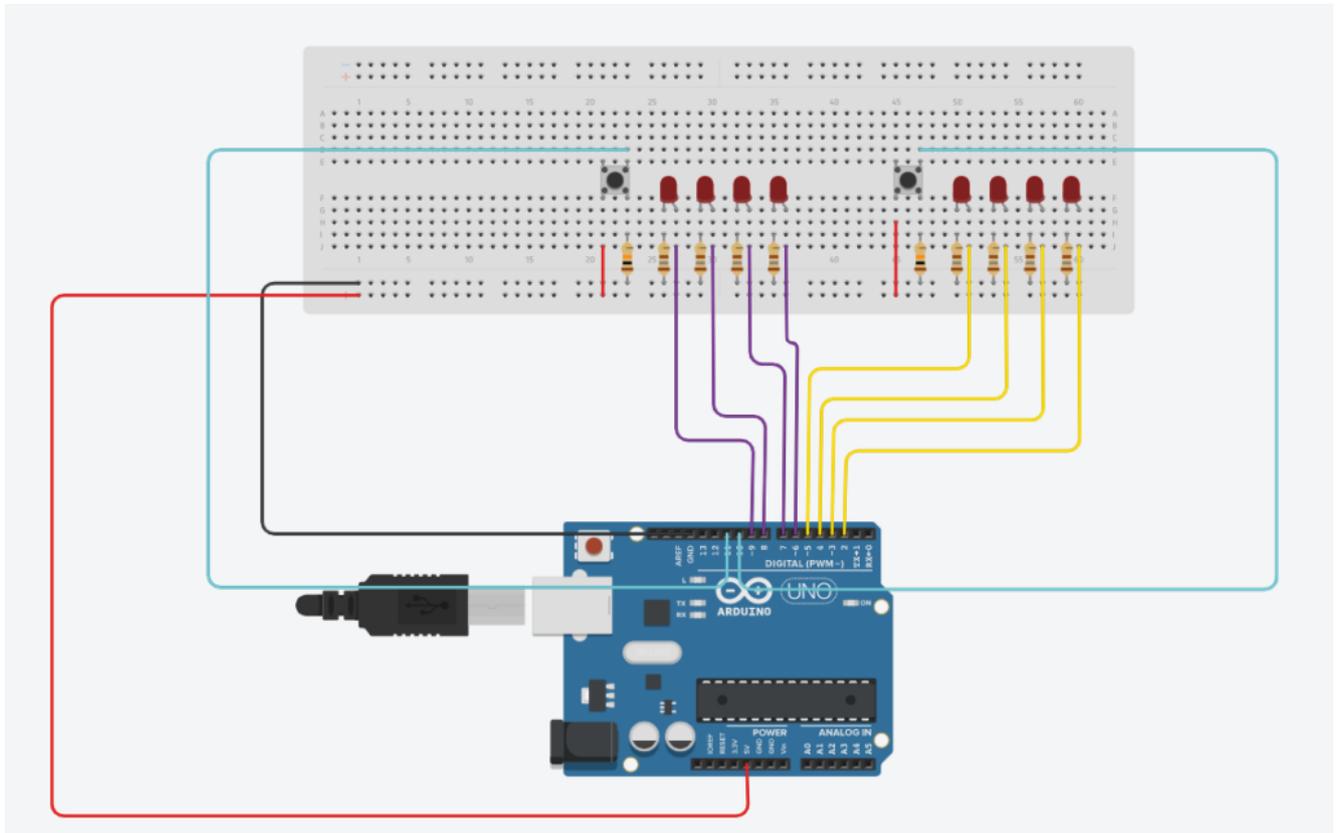
## **Pre-requisiti:**

### [\*LED e Pulsante\*](#)

**Teoria:** Come abbiamo detto, lo scopo dell'esercitazione è quello di attivare due barre LED tramite la pressione di due pulsanti. Le barre simulano l'effetto freccia presente nelle auto Audi. La pressione del pulsante dà il comando di avvio della sequenza e, se si tiene premuto il pulsante, la sequenza rimane attiva.

Sarà necessario sistemare il codice in base a come viene montato il pulsante (se in pull-up o in pull-down)

### Collegamento Circuitale:



### Codice:

Il codice è basato sulla lettura del segnale del pulsante (nel nostro caso in pull-down) che, se premuto, attiva la relativa sequenza

*Esperienza realizzata dalla classe 3 BMC del Dipartimento di Meccanica dell'ITIS "E.Mattei" di Urbino nell'AS 2022-23.*

# Il Cestino Smart

**Obiettivo:** Realizzare un cestino intelligente che si apre quando ci si avvicina, evitando toccarlo.

## Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Sensore Ultrasuoni (e.g., HC-SR04)
- 1 Servomotore (e.g. SG90)
- 1 Scatolone alto
- Colla a caldo

## Pre-requisiti:

[Il Servomotore](#)

[Il Sensore a Ultrasuoni](#)

**Teoria:** Il servomotore è un particolare tipo di motore che consente il controllo della posizione grazie alla componentistica elettronica che ha a bordo. Ad esempio è possibile attivare il servomotore e fermarlo ad un determinato angolo. Alcuni modelli, più avanzati di quello presente in questo esercizio, hanno la possibilità di controllare anche la velocità di rotazione. Il modello di questa esperienza non consente il controllo della velocità, e il suo asse fa una rotazione di 180 gradi.

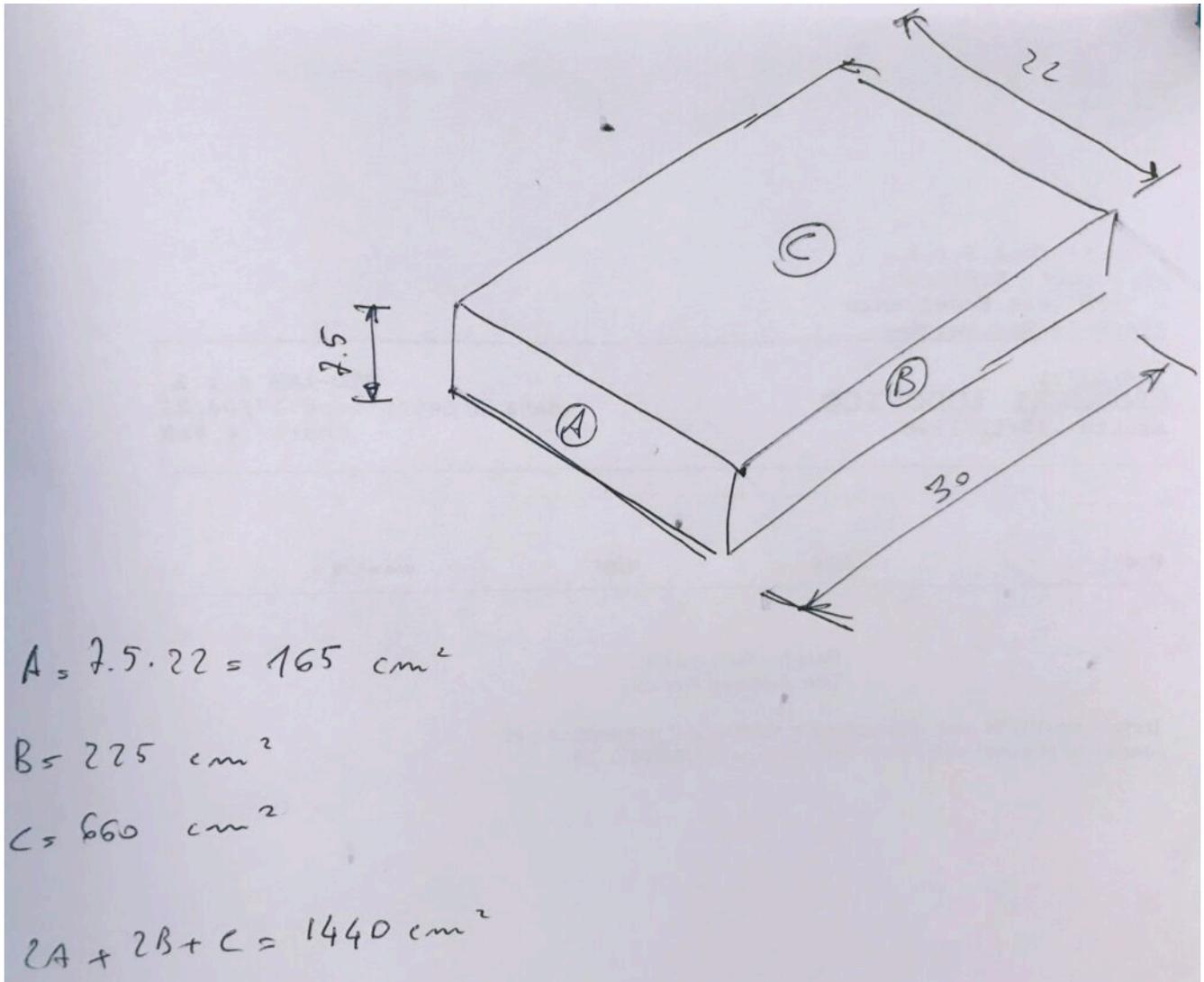
### [SG90-TowerPro](#)

Scheda tecnica del servomotore SG90

Il contenitore del cestino viene realizzato con una scatola, di quelle che contengono le risme di carta A4.



Il coperchio della scatola, pesa complessivamente 61 g ma sarà necessario eliminare i bordi in modo che sia più facile spostarlo con il motore.



Facendo qualche calcolo si ricava che la superficie complessiva del coperchio della scatola è 1440 cm<sup>2</sup>. La densità, quindi è di circa 0,042 g/cm<sup>2</sup> e il peso della sola superficie superiore è

$$0.042 * 600 = 25 \text{ g (circa)}$$

Se a questo aggiungiamo gli elementi che servono per

sorreggere la struttura, possiamo pensare di arrivare ad un peso di 50 g.

Considerando che il peso della superficie superiore della scatola sia uniformemente distribuito, possiamo ipotizzarlo equivalente ad una forza di circa 0.500 N applicata al centro della superficie.

La coppia che il servomotore dovrà sollevare sarà  $(0.5 * 0.15)$

$$M = 0,075 \text{ Nm (oppure } 0,75 \text{ Kg-cm)}$$

Dalla scheda tecnica del servomotore si ricava che esso è in grado di erogare una coppia di 2,5 Kg-cm, ben superiore a quella che ci serve.

Per rilevare la presenza della persona e quindi dare l'ordine alla scheda di aprire il cestino, si utilizzerà un sensore ad ultrasuoni HC-SR04

#### [HC-SR04-ETC](#)

Il sensore di prossimità è un dispositivo che permette di rilevare la presenza di oggetti nelle immediate vicinanze, senza che vi sia un effettivo contatto.

Nel caso specifico, il sensore di prossimità ad ultrasuoni sfrutta il principio del Sonar. Degli impulsi sonori (ultrasonici) vengono emessi dal dispositivo il quale

attraverso l'eventuale eco di ritorno permette di rilevare la presenza di un oggetto all'interno della portata nominale. Esempi pratici di sensori ad ultrasuoni sono i sensori di retromarcia e di parcheggio utilizzati nelle moderne automobili.

Nel dettaglio, le principali caratteristiche tecniche del sensore ad ultrasuoni HC-SR04 sono:

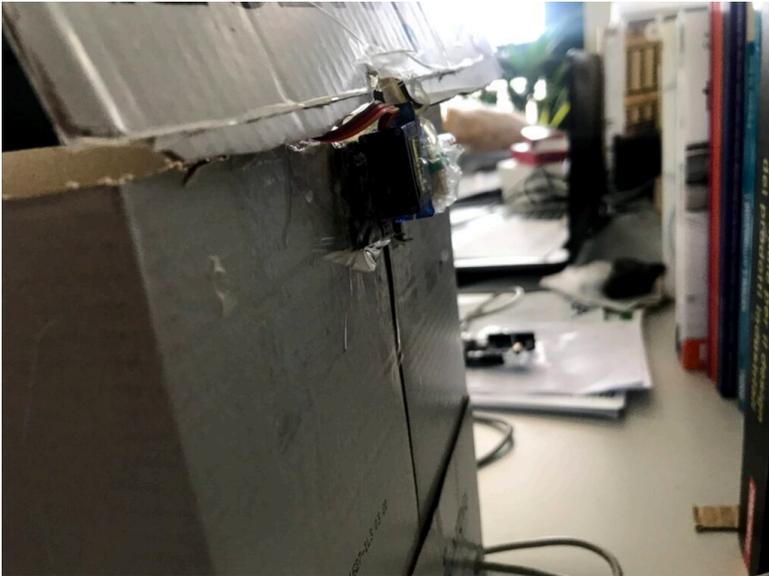
- Alimentazione: +5V DC
- Angolo di misura:  $< 30^\circ$
- Distanza di rilevamento: da 2cm a 400cm
- Risoluzione: 1cm
- Frequenza: 40kHz

**Costruzione del cestino:** dal coperchio tagliare le alette e incollare su una delle superfici un bastoncino di legno (noi abbiamo usato una bacchetta da sushi). il bastoncino deve sporgere leggermente rispetto al piano, per permettere di fissarci il servomotore



Ricavare nella scatola, in uno dei lati corti, uno scanso, per permettere la rotazione del piano con il bastoncino.

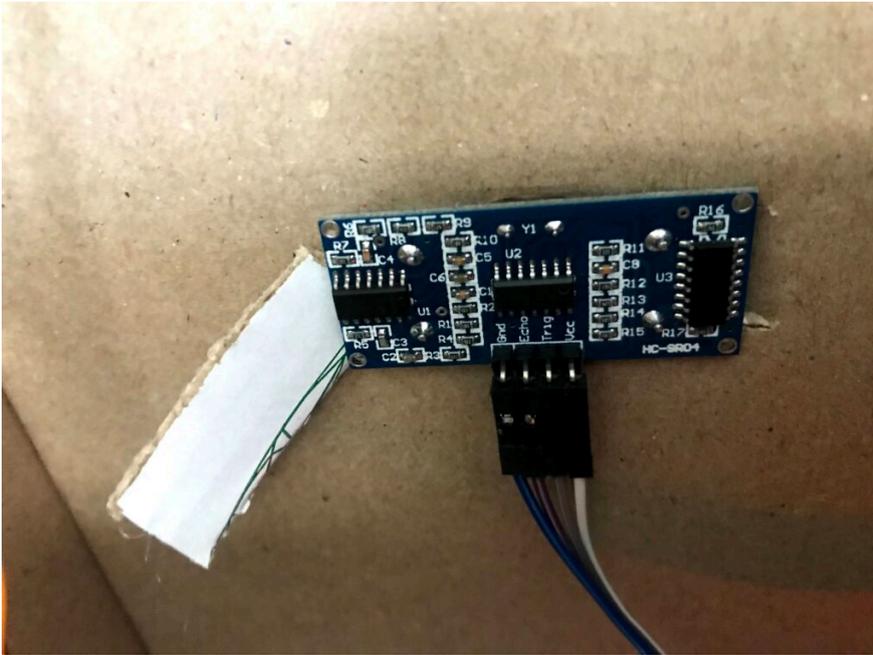
Fissare il servomotore in corrispondenza del taglio, incollandolo sul fianco della scatola. Se serve, si può aggiungere uno spessore sul fianco del servomotore



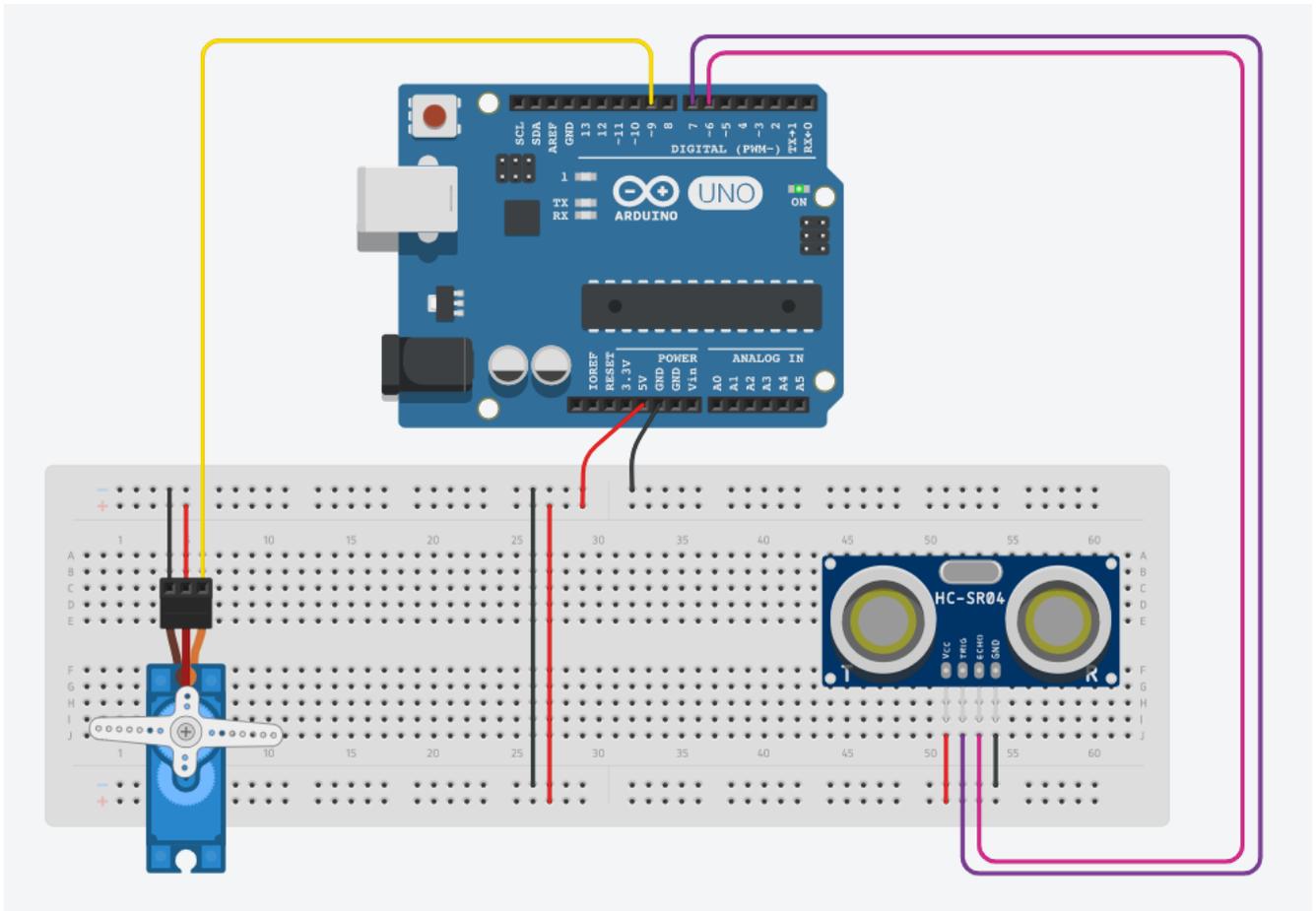
Collegare il servomotore al coperchio superiore, fissando la bacchetta alla levetta del servo



Forare il fianco della scatola in modo da poterci alloggiare il sensore ad ultrasuoni. Il sensore può essere fissato anche ad incastro e senza l'uso di colla o nastro adesivo



**Collegamento Circuitale:**



**Codice:**

Il sensore ad ultrasuoni, rileva la distanza degli ostacoli che ha di fronte. Se il sensore viene posto sul lato del bidone e gli si avvicina una mano, il sensore rileva la presenza dell'arto e invia il segnale al microcontrollore, il quale, elabora il dato e trasmette il movimento di 90 gradi al servomotore al quale è collegato il coperchio del cestino

**Tinkercad:**

**Personalizzazioni:** E' possibile inserire un altro sensore interno al cestino per segnalare quando è pieno

---

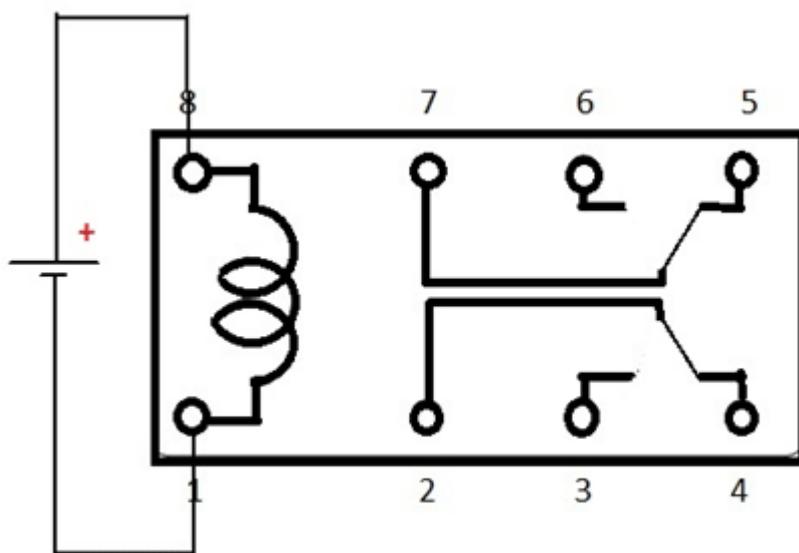
## **Inversione di polarità tramite pulsante (Tinkercad)**

**Obiettivo:** Invertire la polarità di un motore tramite pulsante (e relè DPDT)

## Componenti elettronici:

- Arduino
- Relè DPDT
- Pulsante
- Motore DC 5V

**Teoria:** Uno dei problemi frequenti che si incontra quando si prova a pilotare un motore elettrico è quello di invertirgli il senso di marcia. In commercio esistono molte tipologie di schede che permettono facilmente di controllare il senso di rotazione e la velocità di un motore a corrente continua. La maggior parte di esse monta uno o più relè DPDT, il cui funzionamento è illustrato in figura:

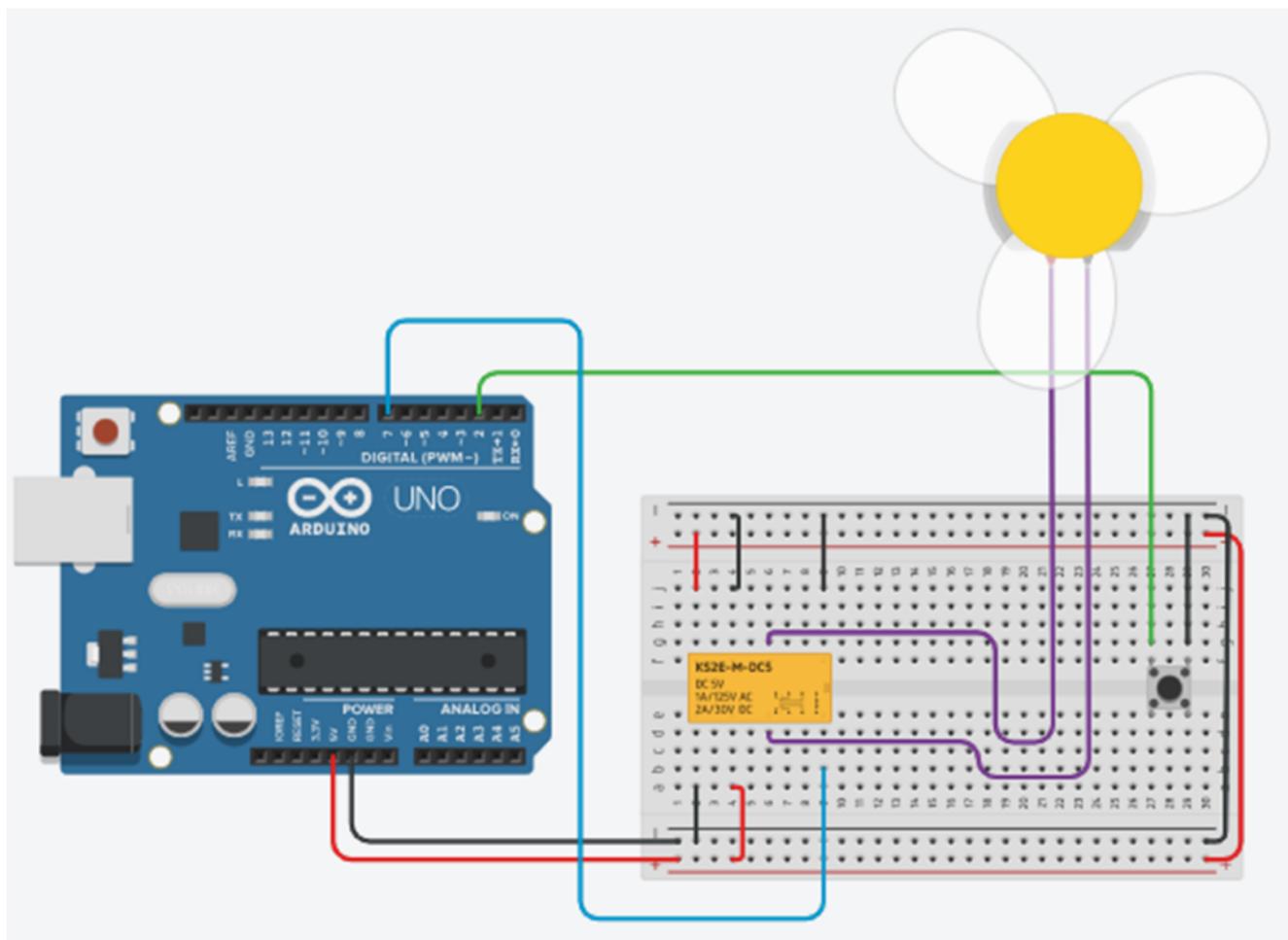


Schema elettrico di un relè DPDT

Una volta eccitata la bobina, il relè apre il contatto NC tra i pin 6-7 e 3-2 e apre quelli NA 5-7 e 4-2. I contatti 7 e 2 sono i COMUNI.

Nell'esempio trattato in questo articolo, un pulsante controlla l'impulso dato alla bobina e, una volta cliccato, la eccita commutando il relè.

### Collegamento Circuitale:



### Codice:

A seguire viene riportato il codice utilizzato:

## Osservazioni:

1. Nel codice il pulsante è stato collegato al pin2 dichiarato come INPUT\_PULLUP. Questa istruzione attiva sul pin la resistenza interna necessaria al funzionamento del pulsante.
  2. Quello che succede è che, una volta premuto il pulsante, i collegamenti in viola al motore cambiano la polarità: il collegamento disposto più in alto passa da negativo a positivo, mentre quello in basso passa da positivo a negativo.
- 

# Controllo di Temperatura e Umidità mediante DHT11

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

## **Componenti elettronici:**

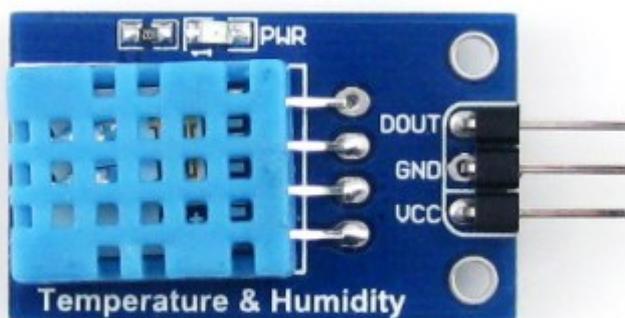
- Arduino UNO
- DHT11

**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.

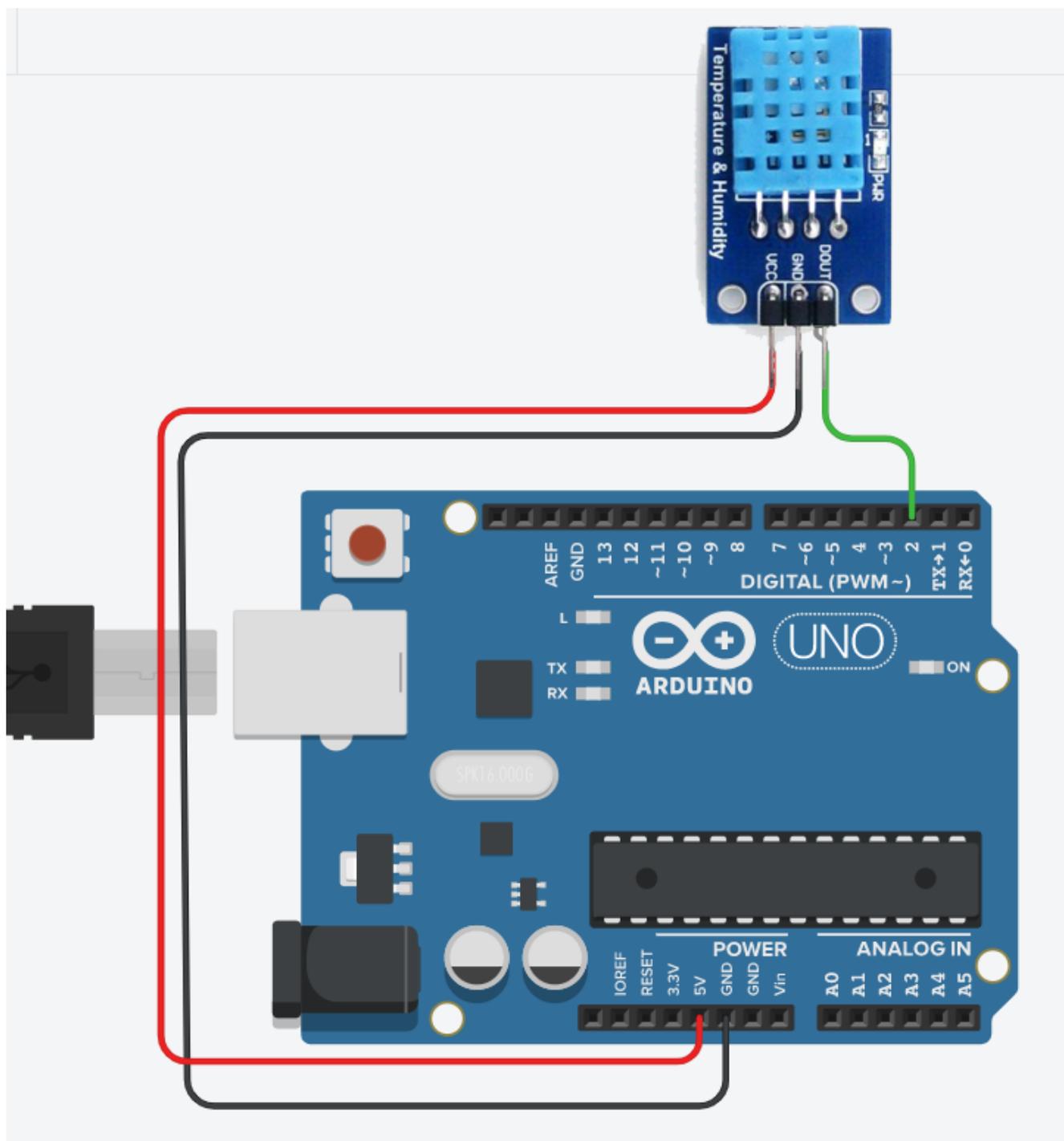


Guardando il dispositivo dal lato della parte sensibile, si ha che il pin di sinistra è quello dell'alimentazione (5V), quello centrale è il GND e il destro è il PIN del segnale digitale, da collegare ad un'uscita digitale di Arduino.

### **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.

---

# **Il Motore passo-passo (Stepper)**

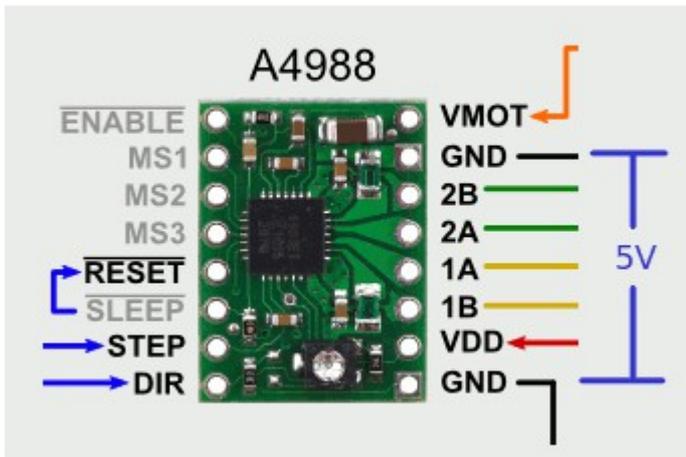
**Obiettivo:** Pilotare un motore passo-passo tramite Arduino

## 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.

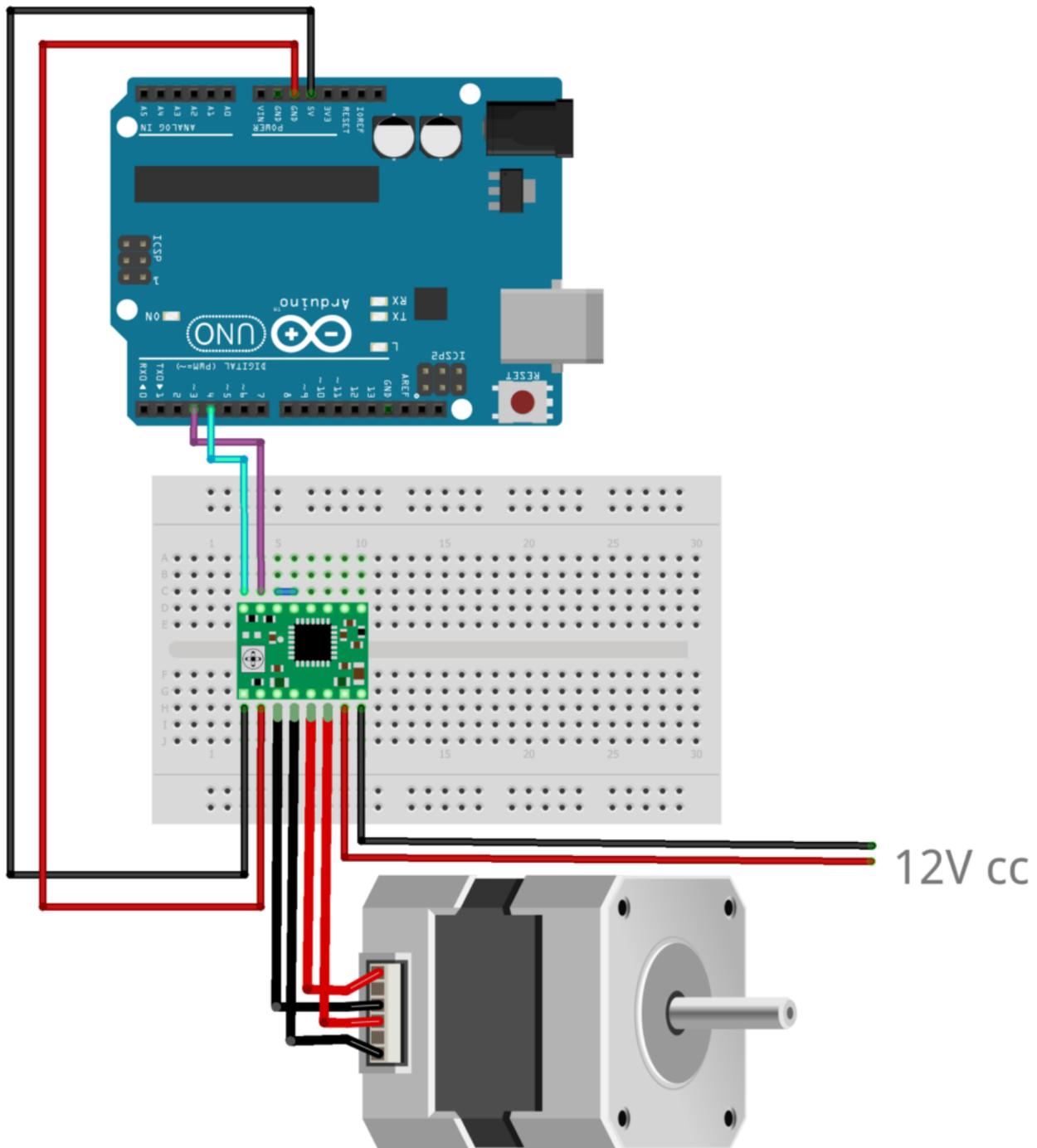
#### [4988-DatasheetScarica](#)

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:



fritzing

### Codice:

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

---

# Azionare un Motore a CC con il Transistor

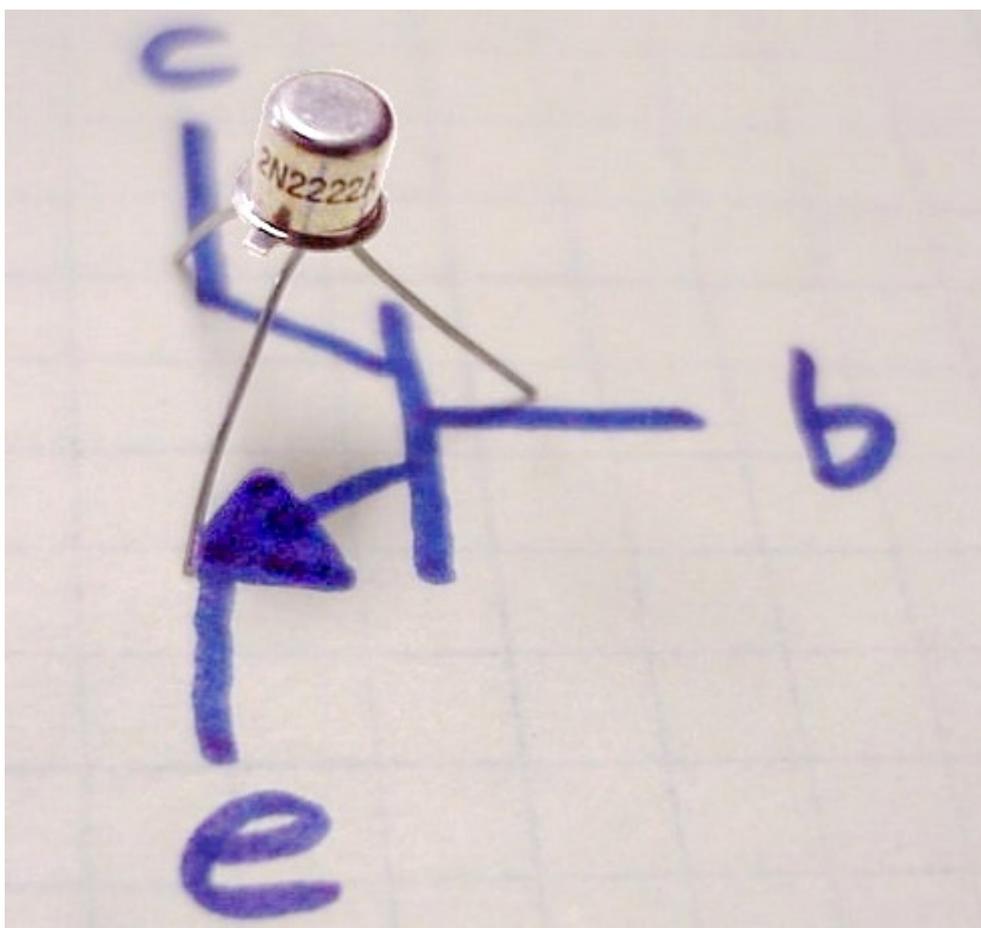
**Obiettivo:** Azionare un motore a corrente continua tramite transistor

## **Componenti elettronici:**

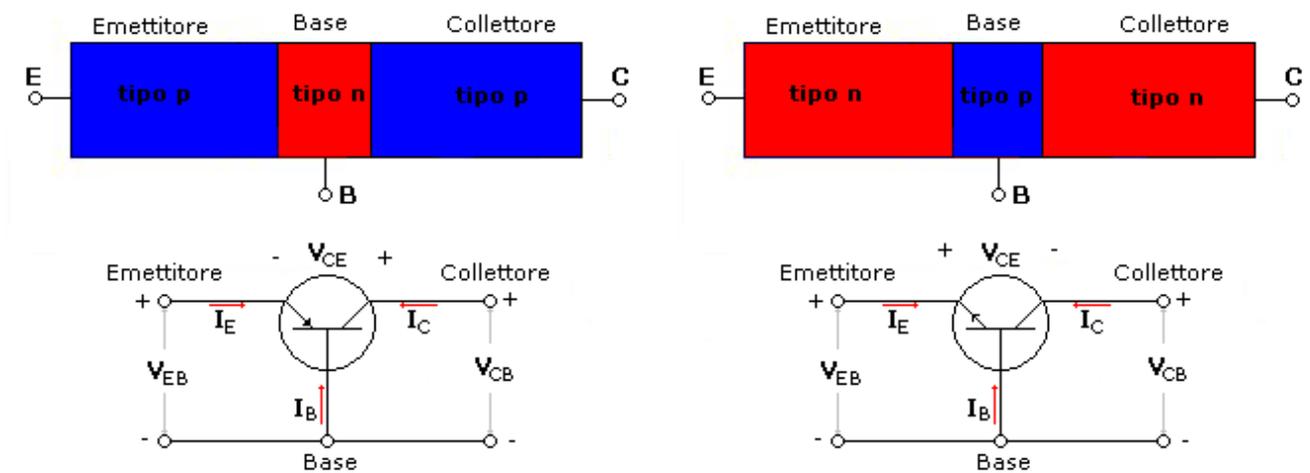
- Arduino UNO
- Breadboard
- 1 [Transistor p2n2222a](#)
- 1 Motore a DC 3V – 6V 17000 RPM
- 1 Resistenza da 1K0hm

**Teoria:** Il Transistor è un elemento elettronico utilizzato come amplificatore di corrente o interruttore. E' costruito da tre strati di materiale semiconduttori uniti con una doppia giunzione p-n, tipica dei diodi. Ad ogni strato è collegato un

terminale: quello centrale si chiama *Base*, e quelli esterni *Emettitore* e *Collettore*. Il principio di funzionamento è basato sulla possibilità di controllare il passaggio di corrente tra collettore ed emettitore, tramite un impulso elettrico fornito alla base.



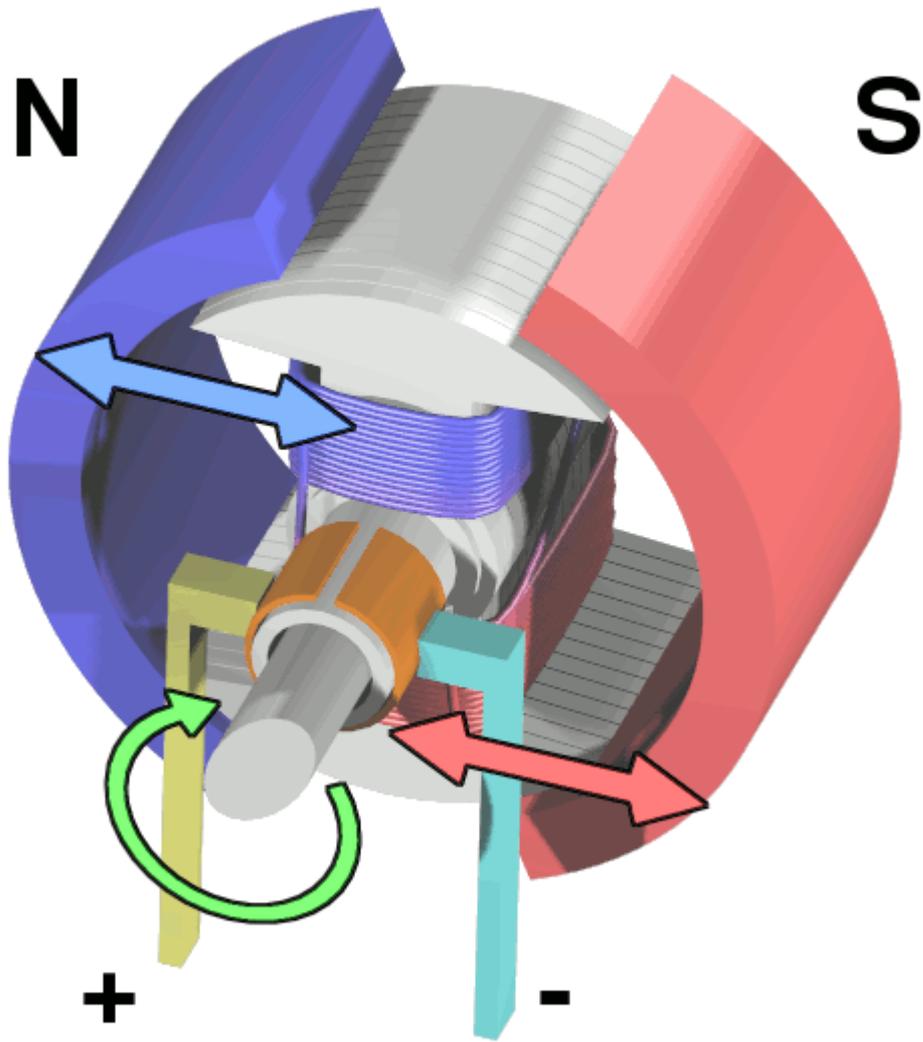
Esistono due tipologie di transistor, a seconda di come sono costruiti: Transistor PNP e NPN; l'unica differenza funzionale tra un transistor PNP e un transistor NPN è la polarità delle giunzioni durante il funzionamento



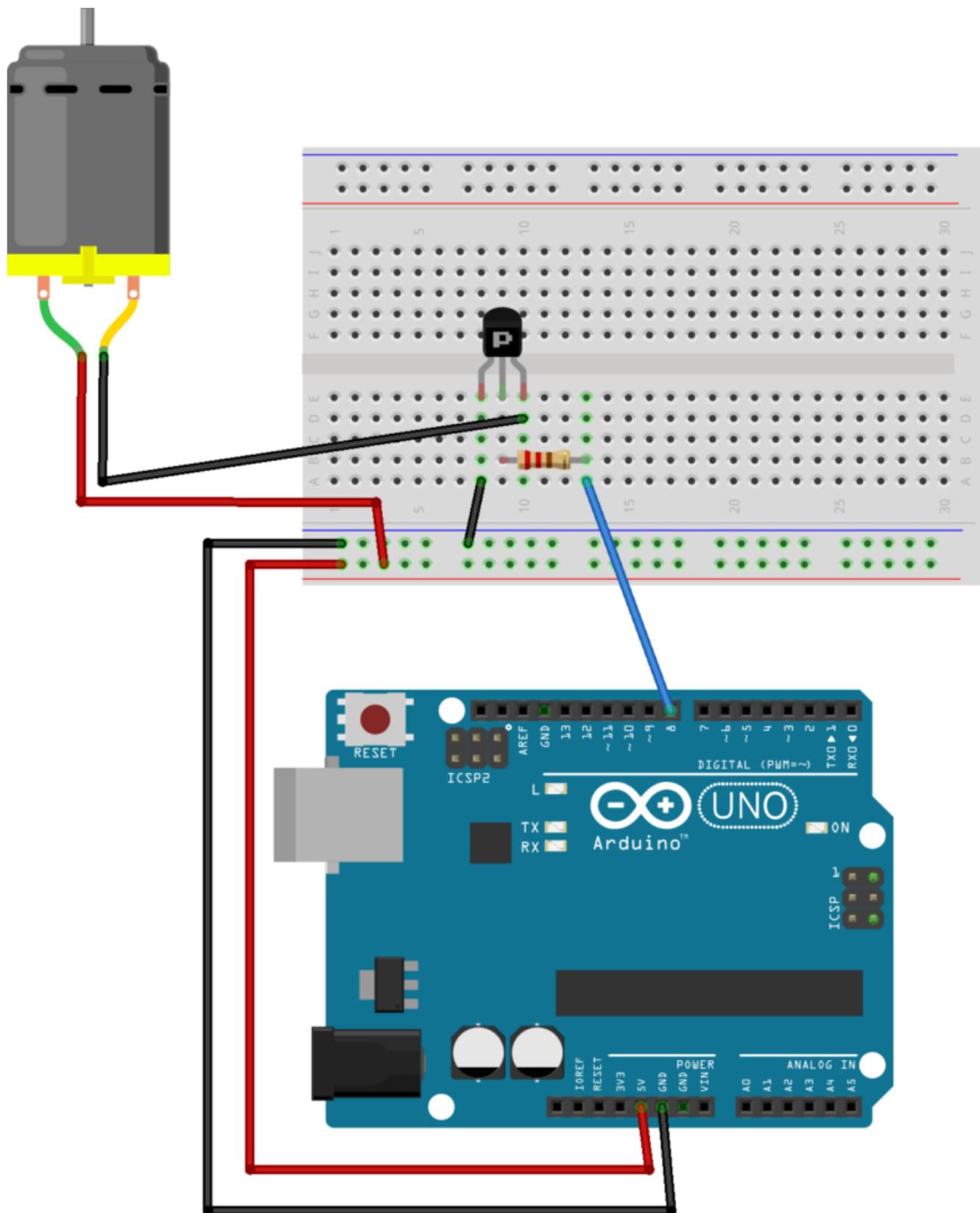
Transistor bipolare a giunzione tipo p-n-p e n-p-n con rappresentazione circuitale.

Tramite i transistor è possibile controllare attuatori che necessitano di grandi correnti, utilizzando una piccola tensione sulla base. In pratica basta collegare un ramo della corrente (ad esempio la terra) sul collettore e sull'emettitore del transistor e gestire il collegamento tramite la base.

**Motori a Corrente Continua (DC):** sono costituiti, al loro interno, da un magnete permanente e da un'elettrocalamita che viene alimentata da dei contatti striscianti, che, per come sono montanti, invertono la tonalità dell'elettrocalamita ad ogni mezzo giro, mantenendo in rotazione l'asse.



Collegamento Circuitale:



fritzing

**Codice:**

**Personalizzazioni:** E' possibile collegare un sensore di temperatura per far attivare il motore quando l'ambiente si surriscalda.

---

## Blinking led

**Obiettivo:** Realizzazione di un led che lampeggi ad una frequenza specifica (e.g., 1Hz)

### **Componenti elettronici:**

- Arduino UNO
- Led

**Teoria:** Il LED (Light Emitting Diode) o Diodo Emittitore di Luce è un dispositivo elettronico che sfrutta le proprietà di alcuni materiali semiconduttori di emettere fotoni (produrre luce).

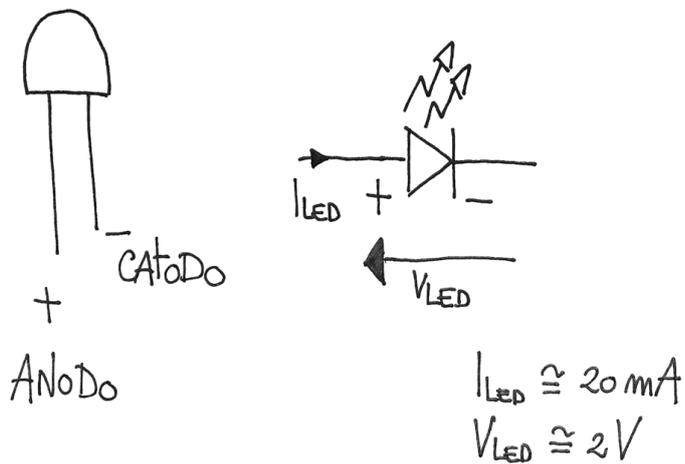
Questo dispositivo è ampiamente utilizzato in molti applicativi realizzati con Arduino ed è caratterizzato da una propria tensione e corrente di funzionamento. Valori tipici sono:

- $I_{Led} = 20 \text{ mA}$

- $V_{LED} = 2 V$

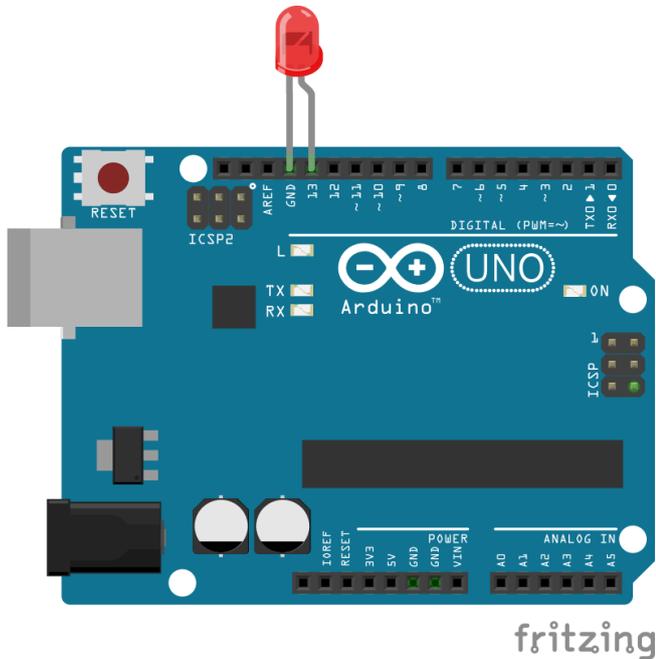
Valori superiori o inferiori possono danneggiare il dispositivo. Per questo motivo si utilizza solitamente una resistenza in serie al LED al fine di limitarne corrente e tensione.

Per un adeguato funzionamento il LED deve anche essere correttamente polarizzato. Nel dettaglio il terminale più lungo di un led rappresenta l'anodo (+) mentre quello più corto il catodo (-).



Rappresentazione grafica di un LED e simbolo circuitale

### Collegamento Circuitale:



## Collegamento Circuitale

### Codice:

L'istruzione **digitalWrite** permette di impostare lo stato logico di un PIN digitale al valore HIGH (5 Volt) o LOW (0 Volt).

Mentre l'istruzione **delay** permette di bloccare Arduino nello stato in considerazione per un certo numero di millisecondi.

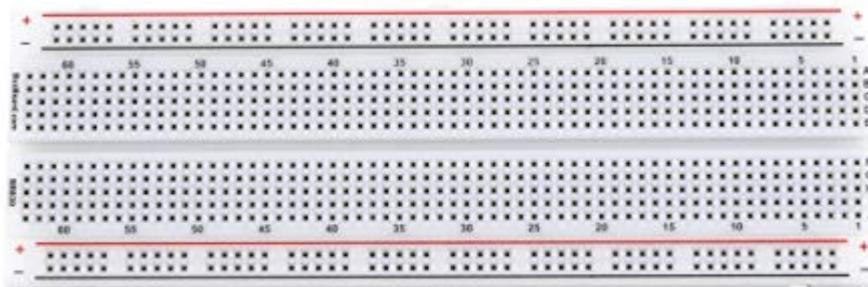
**Personalizzazioni:** E' possibile modificare il comportamento del circuito in questione intervenendo sul valore della variabile *ledTime*. Modificando il suo valore infatti cambia la frequenza di lampeggiamento del LED.

### Approfondimento Teorico:

- Pin13: Nelle prime versioni delle schede Arduino, come riportato nella relativa documentazione: *“there is, however, about 1000 ohms of resistance on pin 13, so you can connect an LED without external resistor”* il PIN13 presenta un’uscita limitata in corrente a causa di una resistenza integrata. Grazie alla presenza di questo elemento non è necessario introdurre una resistenza esterna al fine di limitare la corrente sul LED.
- 

## La Breadboard

**Obiettivo:** Conoscere le caratteristiche e il funzionamento di una breadboard

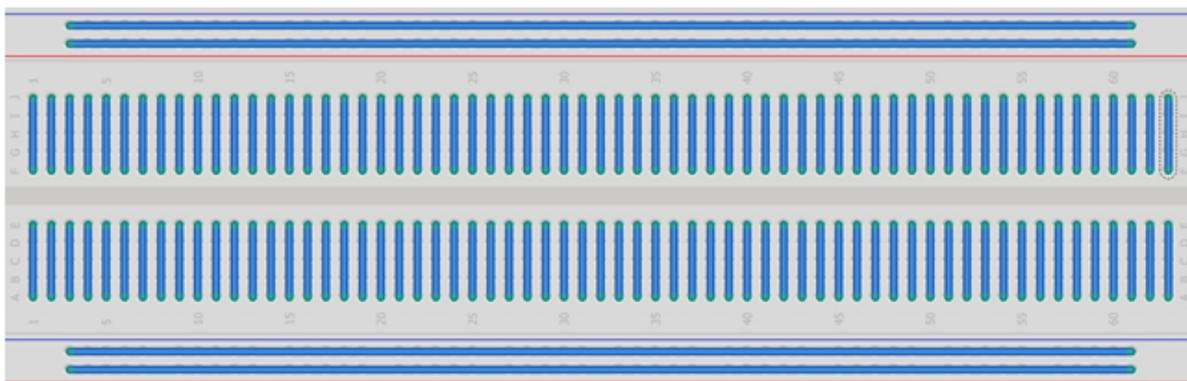


### Teoria:

La breadboard è una basetta che serve a creare prototipi di circuiti elettrici. Ha dei fori su cui vengono inseriti i

cavi. Non necessita di saldature e può essere riutilizzata

Lo schema funzionale della breadboard è il seguente:



Le righe continue rappresentano il collegamento tra i fori.

Le breadboard sono un elemento fondamentale nella costruzione di circuiti con Arduino. Grazie a questo prezioso strumento, è possibile eseguire velocemente i collegamenti tra i vari componenti e moltiplicare le prese di alimentazione e di contatto degli ingressi e uscite della scheda.

Le breadboard sono modulari e possono essere assemblate posizionandole una accanto all'altra.