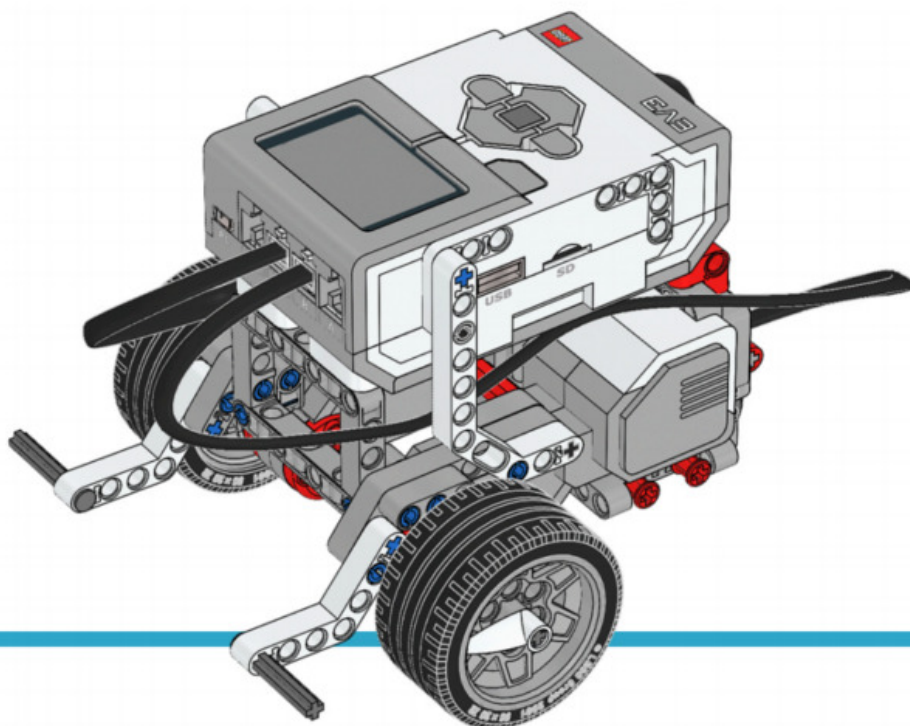


Robot EV3 Mindstorm – Come far curvare un Robot (Driving Base)

Obiettivo: Imparare ad utilizzare i servomotori del Kit EV3 Mindstorm al fine di eseguire traiettorie curve.

Teoria:

I contenuti teorici riportati in questa lezione sono relativi al robot Driving Base EV3. Questo robot è costituito da due servomotori, che permettono di controllare le due ruote, ed una ruota a sfera.

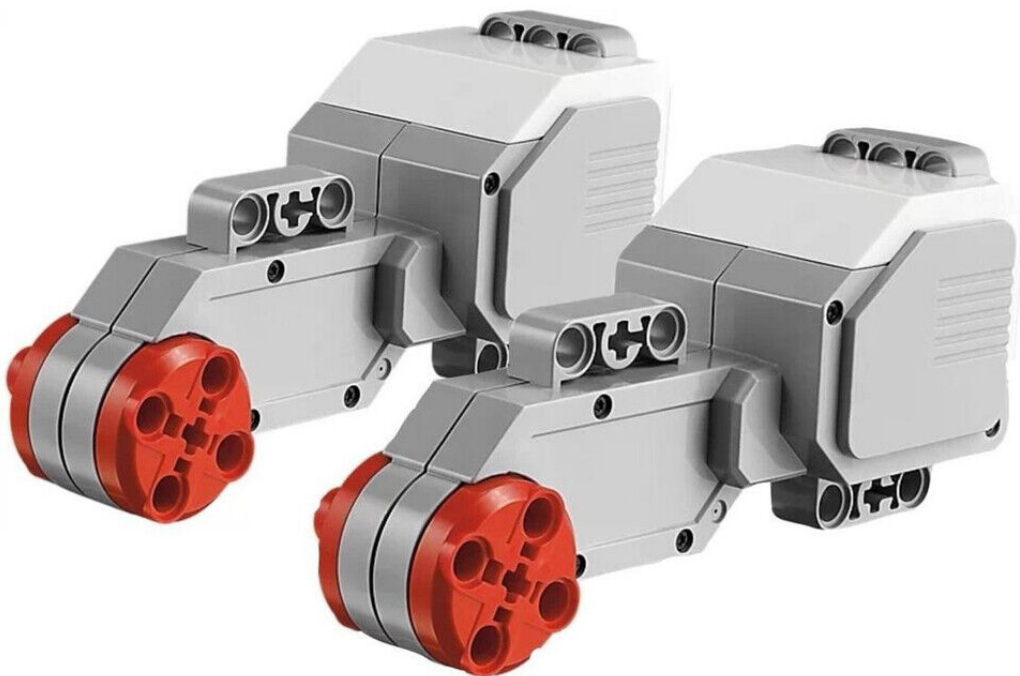


Rappresentazione grafica EV3 Driving Base

I servomotori presenti nel kit EV3 Minstrom presentano le seguenti caratteristiche:

- Contagiri con precisione di un grado
- 160/170 giri al minuto
- Coppia motrice 20 N/cm
- Coppia massima 40 N/cm

MINDSTORMS
education **EV3**

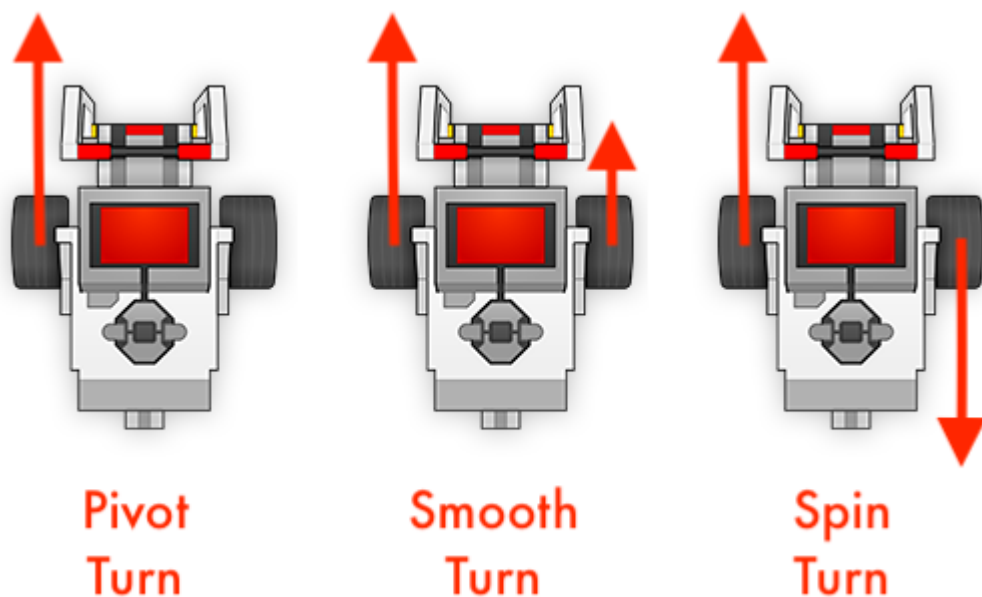


Motori EV3

La modalità di connessione è rapida, è sufficiente connettere i servomotori attraverso l'utilizzo dei cavi al mattoncino EV3 utilizzando le porte di uscita (A, B, C, D). Attraverso la programmazione del mattoncino sarà possibile selezionare la porta impiegata e programmare in modo corretto il servomotore regolando velocità di rotazione e tempo di accensione.

Regolando la velocità di rotazione dei entrambi i servomotori è possibile impostare 3 differenti modalità di curva:

- **Pivot Turn:** una ruota ferma ed una ruota in movimento. In questo caso il robot gira intorno alla ruota ferma, la quale funge da perno. La velocità di rotazione dipende dalla ruota in movimento.
- **Smooth Turn:** le ruote si muovono entrambe nella stessa direzione ma a velocità differente. In questo caso il robot tende a curvare nella direzione della ruota che si muove più lentamente.
- **Spin Turn:** le due ruote si muovono alla stessa velocità ma in direzioni differenti. In questo caso il robot ruota su se stesso. La velocità di rotazione dipende dalla velocità delle due ruote.

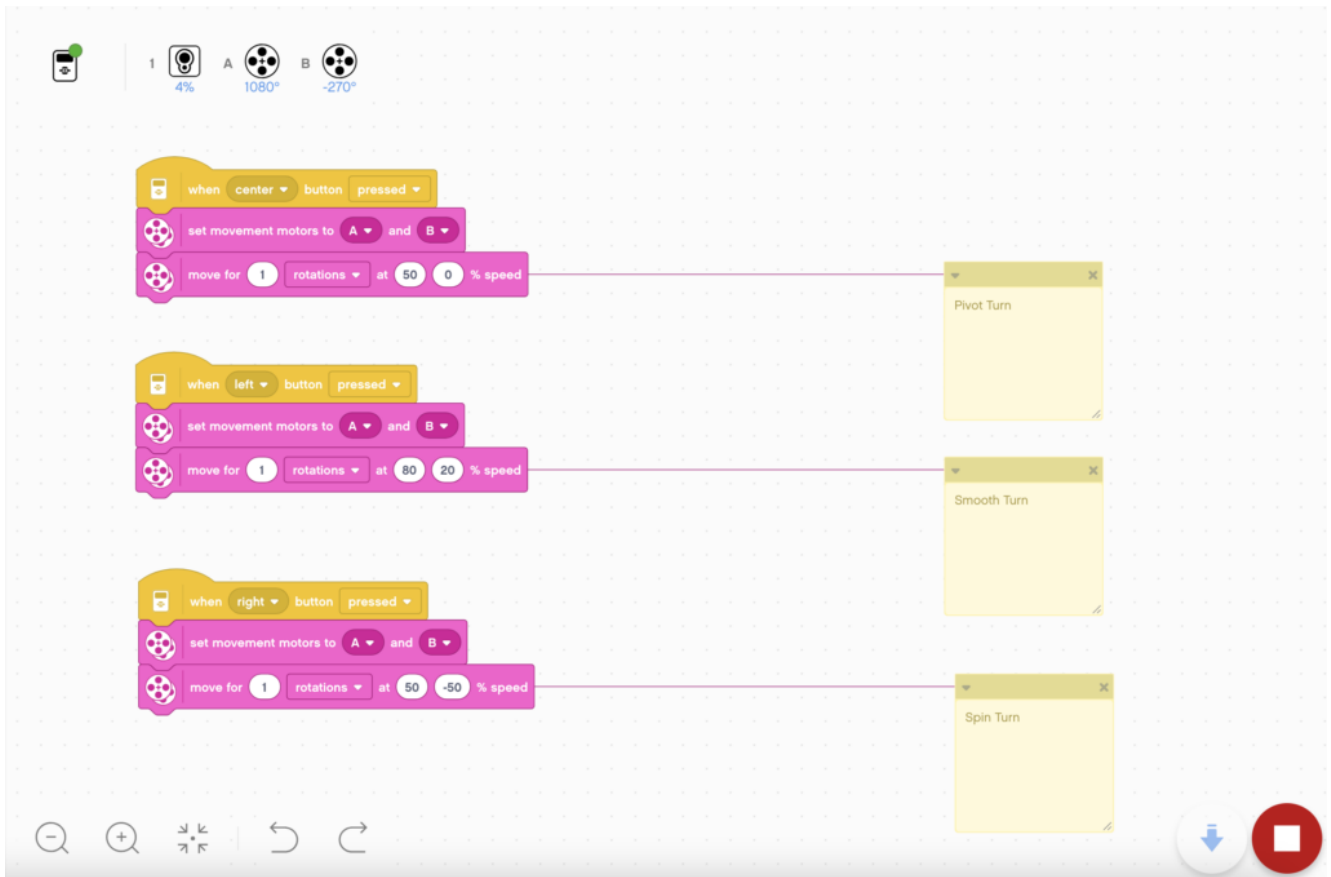


Codice:

Viene in seguito riportato il codice (schema a blocchi) utilizzato per effettuare le tre differenti tipologie di rotazione.

E' importante considerare che attraverso il programma è possibile regolare la **velocità di rotazione** di ogni singola ruota ed il **tempo di rotazione** espresso in: secondi, rotazioni, gradi.

Si potrebbe pensare che attraverso il codice sia possibile definire immediatamente la tipologia di curva desiderata (ad esempio una curva a gomito a 90 gradi), d'altro canto questo non è possibile poiché lo stesso tempo e la stessa velocità di rotazione posso avere effetti differenti su robot differenti.



Codice per fare curvare un robot

- **Pivot Turn:** il programma si avvia quando viene premuto il tasto centrale sul robot. Le ruote effettuano una sola rotazione. Il motore della ruota A è impostato ad una velocità pari al 50% mentre la velocità della ruota B è impostata allo 0%.
- **Smooth Turn:** il programma si avvia quando viene premuto il tasto sinistro sul robot. Le ruote effettuano una sola rotazione. Il motore della ruota A è impostato ad una velocità pari al 80% mentre la velocità della ruota B è impostata allo 20%.
- **Spin Turn:** il programma si avvia quando viene premuto il tasto destro sul robot. Le ruote effettuano una sola rotazione. Il motore della ruota A è impostato ad una velocità pari al 50% mentre la velocità della ruota B è

impostata a -50%.

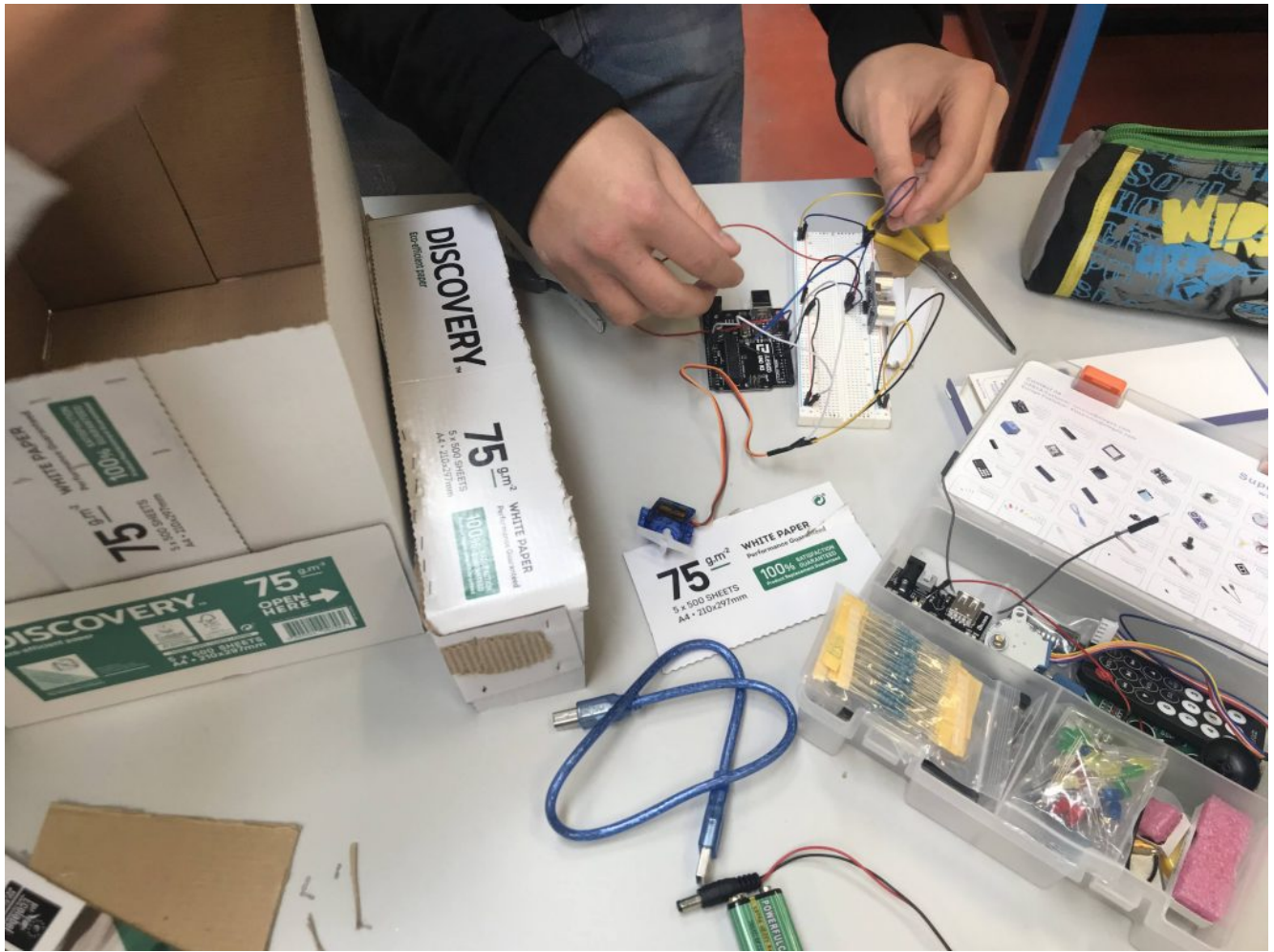
Il Cestino Smart scomposto

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 4×4

Pre-requisiti:

1. Password di accesso con KeyPad 4×4 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;

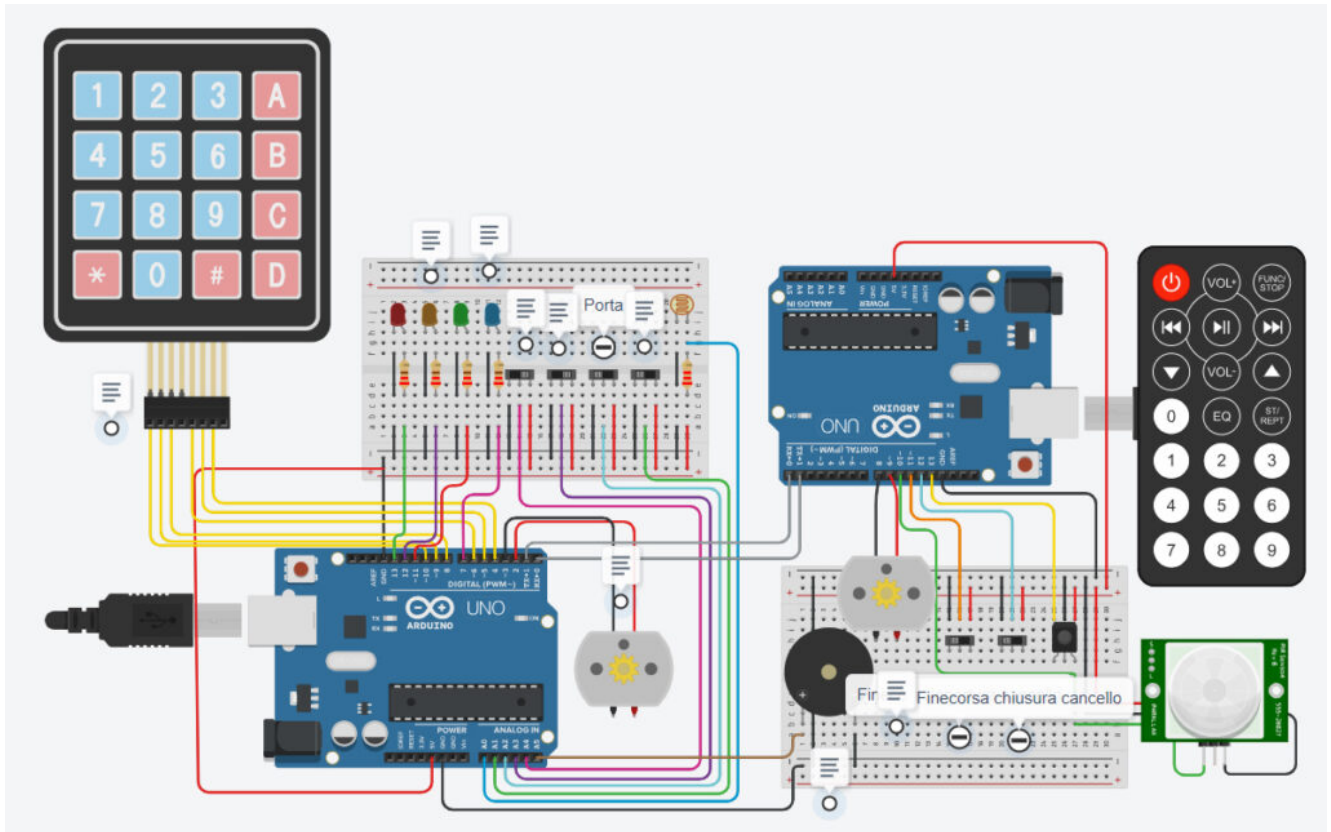
Stato=3 -> Cannello in apertura;

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

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

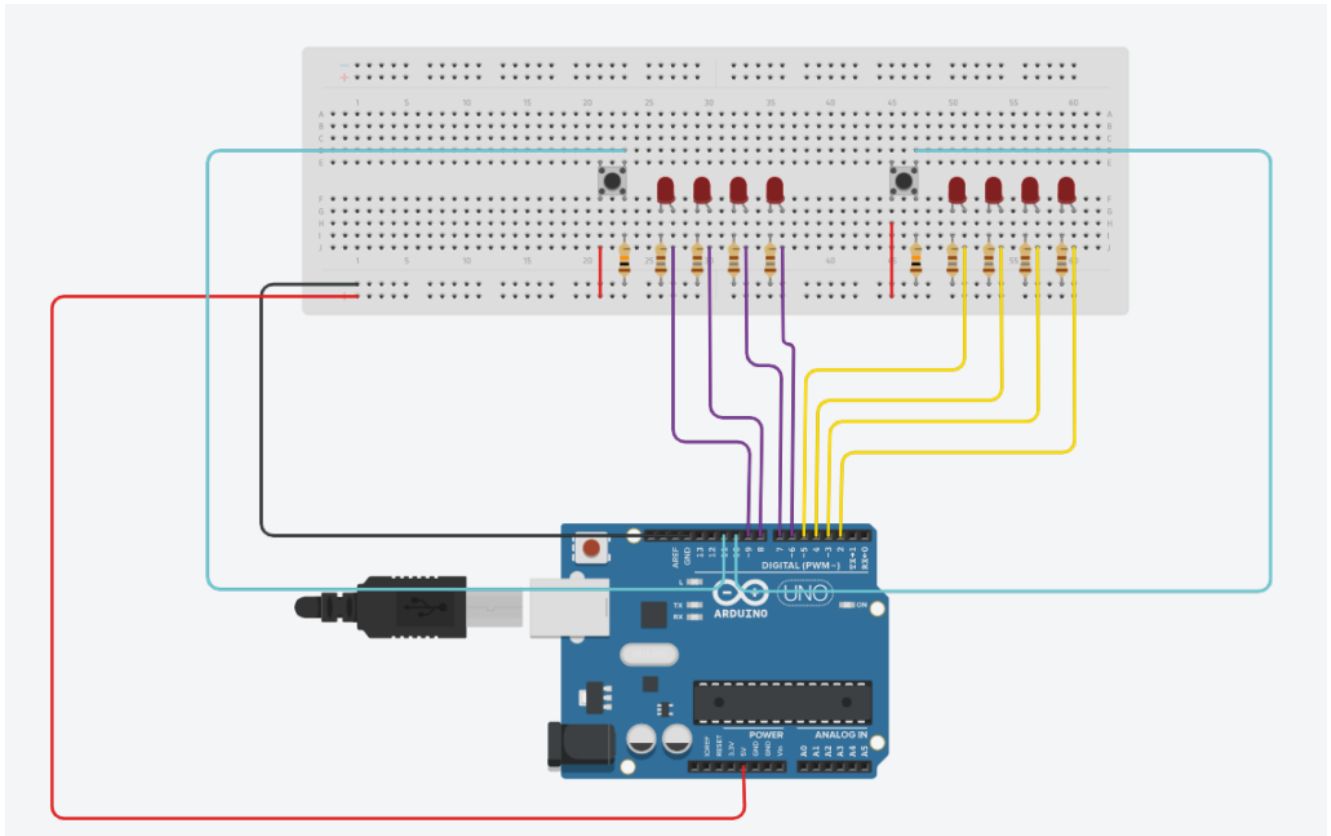
Pre-requisiti:

[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.

Codice e Thinkercad realizzato da Giacomo Brancorsini