

## Scheda esperimento per studente-guida: **FIREBOT AL SOCCORSO**

### - **MATERIALI:**

- Arduino Uno
- Arduino Mega 2650
- Breadboard
- Sensore di colore TCS3200
- Sensore di temperatura TMP36
- Motore con elica
- Stepper Motor
- Batterie da 9V
- Transistor
- Cavi
- Pulsanti
- Resistenze
- Computer
- Cartoncini colorati
- Schermo LCD
- Piastra elettrica che scalda
- Motori a corrente continua
- Arduino "Motor Shield"
- Stampante 3D

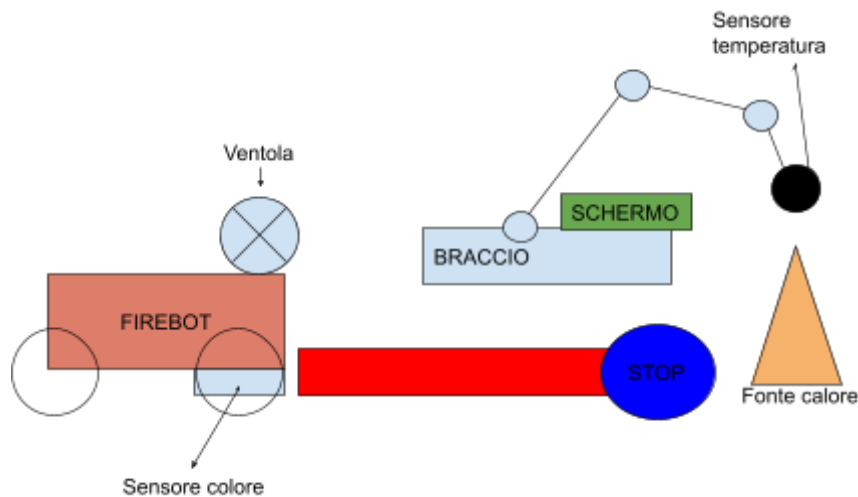
### - **FINALITÀ**

Lo scopo del nostro progetto era simulare un intervento automatizzato per spegnere o prevenire un incendio/surriscaldamento di componenti o attrezzatura. Nel realizzare questo lavoro si può apprendere come utilizzare Arduino e vari sensori, in particolare quello di temperatura e di colore.

### **DESCRIZIONE GENERALE**

Si muove il braccio con i pulsanti per avvicinare il sensore di temperatura posto nella parte anteriore alla piastra precedentemente accesa. Arduino registra dei valori di temperatura superiori a una soglia stabilita; così viene riprodotto un messaggio di avvertenza "ATTENTION HIGH TEMPERATURE!".

A questo punto FIREBOT comincia a muoversi seguendo il rettilineo (nel nostro caso rosso) fino al raggiungimento di una zona di colore differente (per noi blu) che identifica il punto di arrivo. Una volta arrivato la ventola inizia a girare con lo scopo di diminuire la temperatura.



- **FARE** (indicazioni operative)

Per realizzare il lavoro abbiamo diviso il tutto in 3 grandi blocchi:

**BRACCIO ROBOTICO** formato da 3 giunti e collegamenti.

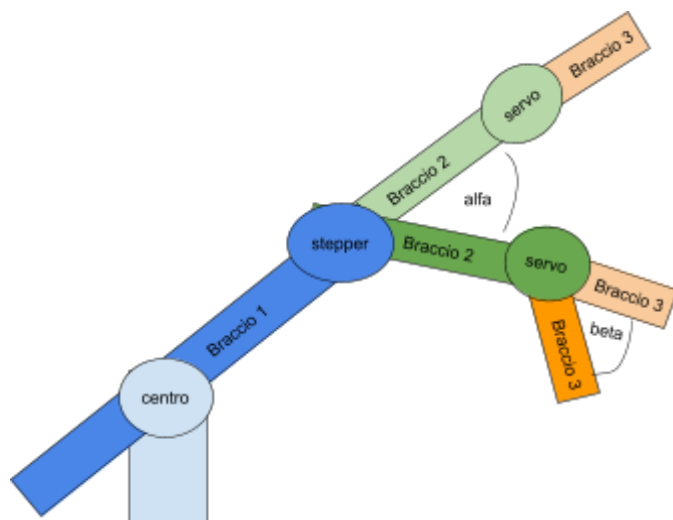
Per creare il corpo del braccio è opportuno progettare in digitale avvalendosi di modellatori 3D, inoltre per evitare eventuali errori di assemblaggio è necessario l'utilizzo di un calibro per prendere le misure in modo preciso.

Per quanto concerne la parte elettronica sono fondamentali gli stepper motor, componenti che permettono il movimento delle varie parti attraverso l'utilizzo di pulsanti. Inoltre è possibile aggiungere al braccio altri sensori, per esempio sull'estremità anteriore è collocato un sensore di temperatura che manda i valori registrati ad Arduino.

Quando la temperatura registrata dal braccio supera una certa soglia, Arduino fa partire un determinato messaggio di avvertenza.

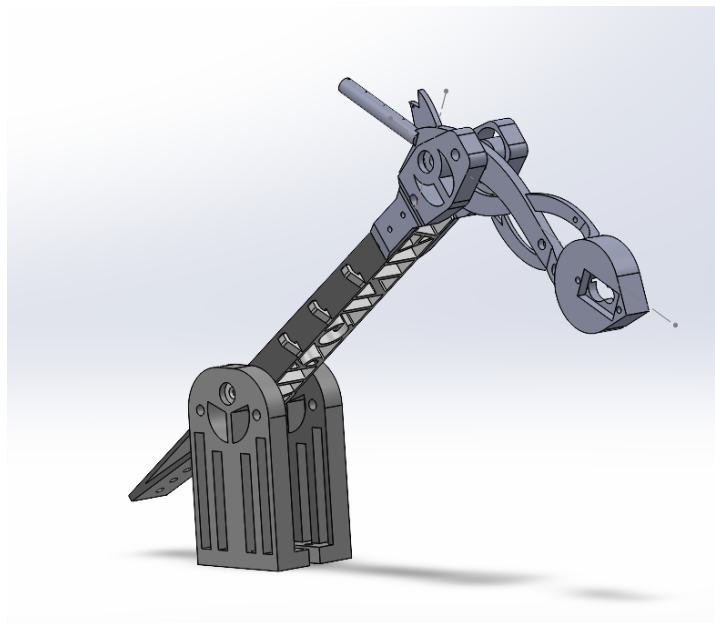
Prestare ATTENZIONE a:

- calcolo dei momenti torcenti: a causa del peso dei motori e della plastica potrebbe essere necessario progettare dei contrappesi per bilanciare e aiutare lo stepper nei movimenti. Qui di sotto riportiamo i nostri calcoli approssimativi.



Elementi	Forza peso (mN)	Braccio (m)	Momenti (mN*m)	Angolo [°]	cos(angolo)	Braccio pieg	Momento pieg
B1	392,4	0,07	27	0	1,0	0,07	27
Stepper	343,35	0,14	48		1,0	0,14	48
<b>Tot m1</b>			<b>76</b>				<b>76</b>
B2	225,63	0,209	47	45	0,7	0,15	33
Servo	127,53	0,278	35	45	0,7	0,20	25
<b>Tot m2</b>			<b>83</b>				<b>58</b>
B3	127,53	0,328	42	30	0,9	0,28	36
<b>Tot m3</b>			<b>42</b>				<b>36</b>
<b>M tot braccio</b>			<b>200</b>				<b>170</b>

→ progettare un elemento alla volta e controllare sul computer che le varie parti si incastrino in modo corretto ed immediato. (Esempio nelle foto sotto riportate per quanto riguarda il braccio)



### **FIREBOT** (unità mobile “firefighter”)

Nella realizzazione di questa unità è **NECESSARIO** iniziare dalla componente elettronica (base arduino e annessi sensori) per evitare problemi nell’assemblaggio (es: spazio insufficiente nella “vettura” e agglomerato di cavi).

Iniziare il progetto attraverso una fase di familiarizzazione con i vari sensori per poi unirli, incorporando uno alla volta i vari codici dei singoli componenti per giungere al progetto finale; così facendo è più facile prevenire e correggere errori nella stesura del codice.

Nel realizzare FIREBOT abbiamo utilizzato in ordine:

- **SENSORE DI COLORE**: per capirne il funzionamento iniziare con un semplice riconoscimento del colore e trascrizione di quest'ultimo su uno schermo LCD.
- **ELICA**: imparare ad attivarla alla pressione di un pulsante.
- **MOTORI A CORRENTE CONTINUA e ESTENSIONE ARDUINO**: per far muovere FIREBOT abbiamo utilizzato due motori a corrente continua controllati dall "Motor Shield" di Arduino. Questa permette di controllare con maggior facilità diversi tipi di motori. Quelli a corrente continua sono più semplici in quanto possono ruotare solo "avanti" o "indietro" a seconda di come colleghiamo il "+" e il "-". La velocità è regolata dalla quantità di corrente collegata, in particolare questi motori hanno bisogno di almeno 9V per funzionare.

Prestare attenzione a:

- Scegliere colori con valori che non si sovrappongono per evitare l'erronea attivazione delle componenti;
- calibrare di volta in volta i valori dei colori poiché variano in base alla luce presente nell'ambiente circostante;
- avvicinare il più possibile il sensore di colore al cartoncino per essere nelle condizioni migliori e permettere una buona rilevazione;

#### 1. **PARTE NON ELETTRONICA** (piastra e assemblaggio del tutto)

Dopo aver costruito un rettilineo a tinta unita da far seguire a FIREBOT, definire ad Arduino i valori RGB del relativo colore.

Inoltre ritagliare un cartoncino di colore differente che rappresenterà la zona dove FIREBOT si dovrà fermare in prossimità della piastra.

Prestare ATTENZIONE a:

- non superare la soglia massima del sensore di temperatura rischiando di danneggiarlo;

### **DOMANDE E RISPOSTE**

#### 1. *Perché per identificare un colore abbiamo 3 valori?*

Ogni colore si può identificare attraverso una serie di numeri, questa è definita dalla sigla RGB (red-green-blue) perchè contiene 3 strati sovrapposti di Rosso, Verde e Blu codificati su 256 livelli ciascuno (da 0 a 255). Per esempio il rosso viene identificato attraverso i numeri (255, 0, 0)

2. *Perché abbiamo utilizzato questo tipo di motore nel braccio?*

Abbiamo utilizzato uno stepper motor perché è più preciso e facile da controllare rispetto a un servo motore.

3. *C'è la possibilità di aggiungere ulteriori sensori e funzioni oltre a quello che FIREBOT sa già fare?*

Sicuramente, ma questo richiederebbe un ulteriore lavoro di approfondimento sui sensori e su come funzionano.

4. *Che applicazioni può avere questo esperimento nella realtà di tutti i giorni?*

Quello che abbiamo creato è un modello per prevenire e spegnere incendi in modo automatizzato, senza l'intervento diretto di umani

## - **INTERPRETAZIONE A LIVELLO AVANZATO**

In questa sezione analizzeremo alcune parti di codice del progetto in modo più approfondito affinché chi fosse interessato abbia la possibilità di comprendere più a fondo il nostro lavoro.

Codice FIREBOT:

```
void loop() {
  digitalWrite(S2, LOW); //in questa parte facciamo in modo che venga attivato il sensore di colore:
  digitalWrite(S3, LOW); //accendendo e spegnendo vari pin arduino è in grado di ricavare i
  r = pulseIn(sensorOut, LOW); //valori del rosso, verde, blu (RGB);
  digitalWrite(S2, HIGH);
  digitalWrite(S3, HIGH);
  g = pulseIn(sensorOut, LOW);
  digitalWrite(S2, LOW);
  digitalWrite(S3, HIGH);
  b = pulseIn(sensorOut, LOW);

  Serial.print("R="); //con questa funzione scriviamo i valori relativi ai colori
  Serial.print(r);
  Serial.print("\tG=");
  Serial.print(g);
  Serial.print("\tB=");
  Serial.println(b);

  delay(100);
  if ( ( r >= 21) && ( g >= 21) && ( b >=20 ) ) { //funzione che permette di attivare
    digitalWrite(motorPin, HIGH); //la ventola (motorPin) con determinati valori RGB (a scelta)
  } else {
    digitalWrite(motorPin, LOW);
  }
  if( (r <= 20) && (g <= 20) && (b <= 17) ) { //funzione che muove lo stepper motor (steve)
    steve.step(100);}
}
```

Codice per i motori del braccio:

```
//stepper
val1 = digitalRead(puls1); //determiniamo il segnale dato dai pulsanti;
val2 = digitalRead(puls2stepper);

if(val1 = HIGH){//se il primo pulsante è premuto lo stepper si muove in avanti;
  //Serial.print(val1)
  stepper.step(70);
}
if(val2 = HIGH){//se il secondo pulsante è premuto lo stepper si muove indietro;
  //Serial.print(val1)
  stepper.step(indietro);
}
```

Codice semplice per far girare i motori a corrente continua con la Shield

```
void setup() {
  pinMode(12, OUTPUT);
  digitalWrite(12, HIGH);
  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);

  analogWrite(3, 255);
  analogWrite(11, 255);
}

void loop() {

}
```

Faudo Virginia, Lazar Loredana, Ligato Alessandra, Locatelli Alessandro