

Air Ice Hockey

Titolo del progetto:	Air Ice Hockey
Alunni:	Alessia Sarak, Giairo Mauro, Joey Biancardi, Alessandro Spagnuolo
Classe:	Info I3
Anno scolastico:	2017/2018
Docenti responsabili:	Luca Muggiasca e Adriano Barchi

1	Introduzione	3
1.1	Informazioni sul progetto	3
1.2	Abstract	3
1.3	Scopo	3
	Analisi	3
1.4	Analisi del dominio	3
1.5	Analisi e specifica dei requisiti	4
1.6	Pianificazione	4
1.7	Analisi dei mezzi	6
1.7.1	Software	6
1.7.2	Hardware	6
2	Implementazione	7
3	Test	15
3.1	Protocollo di test	15
3.2	Mancanze/limitazioni conosciute	15
4	Consuntivo	16
5	Conclusioni	17
5.1	Sviluppi futuri	17
5.2	Considerazioni personali	17
6	Allegati	17

1 Introduzione

1.1 Informazioni sul progetto

Il progetto consiste nel fare un campo da Air Ice Hockey dove l'avversario è il computer. Esso deve saper rispondere all'avversario in tempo reale.

Il gruppo responsabile del progetto è composto da:

- Alessia Sarak della I3AA
- Giairo Mauro della I3BB
- Joey Biancardi della I3BB
- Alessandro Spagnuolo della I3BB

I docenti responsabili del progetto sono Adriano Barchi e Luca Muggiasca.

1.2 Abstract

The purpose of this project is to have an ice air hockey table that function.

This game is composed of two players; one is the computer and one is the person. The Computer part will be programming with Arduino mega.

There is an application that already exist (JJRobots), it will take the data of the table and the disk and, with Wi-Fi connection, it will send all data on our program. The data will be interpreted from the program and then the program will make his move. This move must be instantaneous and the program must decide which strategy use. There are 3 strategies: attack, defence and attack and defence.

The computer will answer in real time.

1.3 Scopo

Lo scopo del progetto è di creare un campo funzionante di Air Ice Hockey. Il gioco è composto da due giocatori: il computer e la persona.

Il computer dev'essere in grado di rispondere ai colpi dell'avversario in tempo reale tramite una webcam che legge i valori del disco e calcola la direzione in cui andrà, in questo modo riesce a capire dove muovere il robot, in modo che sia del tutto autonomo una volta fatta partire l'applicazione.

Analisi

1.4 Analisi del dominio

Abbiamo un campo da air hockey classico per 2 giocatori.

Quello che dobbiamo creare è già stato creato come progetto, JJRobots, e noi dobbiamo replicarlo usando i pezzi che i professori ci hanno dato.

L'idea del progetto è presentarlo a espoprofessionisti agli interessati alla nostra scuola e in particolare alla nostra sezione (Scuola Arti e Mestieri di Trevano, sezione informatici).

Ci sarà un solo giocatore che si sfida con un sistema robotizzato che si muove grazie ad Arduino. Il come è spiegato nella sezione "scopo" e nel capitolo "implementazione".

Per poter implementare questo progetto bisogna avere delle buone conoscenze su Arduino e bisogna conoscere come far funzionare i motori con la scheda di JJRobots per poi poterle applicare alla programmazione.

1.5 Analisi e specifica dei requisiti

ID	REQ-001
Nome	Tavolo
Priorità	1
Versione	1.0
Note	-
Sub-ID	Sviluppo
001	I componenti che muovono il robot sono montati e funzionali
002	Accanto al punteggio c'è il segnatore di tempo rimanente

ID	REQ-002
Nome	Strategie di gioco
Priorità	1
Versione	1.0
Note	-
Sub-ID	Programmazione
001	Nel programma devono essere presenti 3 strategie di gioco
002	Strategia 1: Difesa
003	Strategia 2: Attacco
004	Strategia 3: Difesa - Attacco

ID	REQ-003
Nome	Movimento computer
Priorità	1
Versione	1.0
Note	-
Sub-ID	Programmazione
001	Il telefono che inquadra il campo da gioco deve essere collegato con il programma
002	Vengono registrati i dati del disco in movimento
003	I dati vengono utilizzati per definire quale strategia di gioco utilizzare
004	Viene attivata la strategia di gioco

1.6 Pianificazione

Il Gantt è suddiviso in giorni e ogni giorno è diviso in tre: le prime 2 ore, le seconde 2 ore e le terze 2 ore. Ogni colore rappresenta un membro del gruppo; il rosa rappresenta Alessia Sarak, il verde rappresenta Joey Biancardi, il blu rappresenta Giairo Mauro e il giallo rappresenta Alessandro Spagnuolo. Il 19 gennaio è libero nel caso di qualche ritardo o nel caso si volesse perfezionare qualcosa.

Descrizione	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
	10.nov			17.nov			24.nov			01.dic			15.dic			22.dic			12.gen			19.gen		
Analisi																								
Raccolta informazioni																								
Analisi Costi + Benefici																								
Analisi Requisiti																								
Progettazione																								
Gantt																								
Implementazione																								
Ricerca di informazioni riguardo l'applicazione JJRobots																								
Stampa 3D dei pezzi																								
Studio del tavolo e dei pezzi																								
Raccolgimento dati																								
Algoritmo di punteggio																								
Algoritmo di timer																								
Programma di difesa																								
Programma di attacco																								
Unire attacco e difesa																								
Unire tutti i programmi																								
Test																								
Test algoritmo di punteggio																								
Test algoritmo di timer																								
Test algoritmo di difesa																								
Test algoritmo di attacco																								
Test algoritmo di attacco e difesa																								
Test che tutto funziona																								
Documentazione																								

Viene aggiornata durante tutto il corso del progetto

1.7 Analisi dei mezzi

A disposizione c'è un tavolo per il gioco, la stampante 3D per stampare i pezzi da montare sul tavolo, l'applicazione JJRobots per prendere le misure del tavolo, una 7 segmenti Adafruit per segnalare il tempo alla fine del gioco e il nostro portatile.

1.7.1 Software

- Arduino IDE (programma)
- AHR Robot (JJRobots)
- Word (documentazione e diari)
- Excel (Gannt)
- Google Chrome (come visualizzatore di pdf e per le ricerche)
- PowerPoint (presentazione)
- Fritzing (schema montaggio di arduino)

1.7.2 Hardware

Il prodotto verrà sviluppato su un campo da Air Ice Hockey. La base è identica a quella di un campo normale, la differenza sta nel campo dell'avversario che, essendo computerizzato, ha un pusher montato su delle sbarre, esse sono collegate a delle cinghie. Per permettere il movimento del pusher sono necessari anche la scheda di JJRobots, 2 motori passo-passo, e per definire la direzione del movimento si necessita di un telefono (sistema operativo android) che tramite l'applicazione di JJRobots (Air Hockey) riesce a prevedere il movimento che verrà fatto dal puck.

Lista dei pezzi utilizzati (per vedere come montarli tra di loro andare sotto implementazione, pagina 7):

- Tavolo
 - 1 – Tavolo da Air Ice Hockey
 - 2 – Pusher
 - 1 – Puck
 - 2 – Sbarre
 - 1 – Cinghia
 - 2 – Motori passo-passo
 - 1 – Telefono (sistema operativo Android) con su l'applicazione di JJRobots (Air Hockey)
- Timer
 - 1 – Arduino Uno
 - 1 – 7 segmenti Adafruit
 - 2 – Resistenze
 - 1 – Led Blu
 - 1 – Bottone
- Programma con Arduino
 - Arduino Leonardo

2 Implementazione

Per poter fare il tavolo di Air Ice Hockey funzionante il lavoro è stato suddiviso in 3 parti: motori, algoritmo di timer e montaggio del tavolo, rispettivamente ci hanno lavorato Giairo Mauro e Alessandro Spagnuolo, Alessia Sarak, Joey Biancardi.

Motori

Componenti utilizzati

- 1 Arduino Mega (serve Arduino Leonard)
- 2 motori Stepper Motor
- 2 A4988 Stepper Motor Driver
- 1 Scheda JJRobots

Montaggio

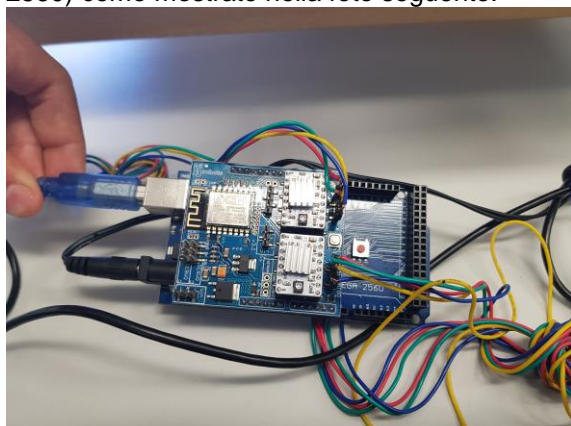
Abbiamo montato un radiatore su ogni Driver per evitare il riscaldamento di essi, una volta fatto abbiamo collegato i due driver alla scheda JJRobots.



I cavi del motore sono stati saldati a dei componenti e poi collegati alla scheda, i cavi sono stati saldati secondo un ordine preciso mostrato nel Datasheet dei motori.



Una volta finito il montaggio della scheda JJRobots l'abbiamo collegata all'arduino fornitoci (arduino mega 2560) come mostrato nella foto seguente.



Test motori

Attributi:

- `const int enaM1M2Pin = 4;` → pin per l'abilitazione dei due motori
- `const int microStepPin = A4;` → pin per suddivisione dei passi in micro passi
- `const int pushButtonPin = A3;` → pin per il Pulsante sul board
- `int Val = 0;` → variabile per la lettura pulsante

- `const int stepM1Pin = 7;` → Pin per step motore 1
- `const int dirM1Pin = 8;` → Pin per direzione motore 1

- `const int stepM2Pin = 12;` → Pin per step motore 2
- `const int dirM2Pin = 5;` → Pin per direzione motore 2

Il programma fa fare un giro in avanti e uno indietro ai motori, ma se si preme il bottone presente sulla scheda JJRobots fa solo un giro in una direzione.

`digitalWrite(dirM1Pin, HIGH);` → Inizialmente si abilita la direzione in cui si vogliono muovere i motori con questo comando: con "HIGH" si muove in avanti con "LOW" si muove indietro.

```
if (digitalRead(pushButtonPin) == HIGH) {
  for(int x = 0; x < 3200; x++) {
    digitalWrite(stepM1Pin, HIGH);
    delayMicroseconds(80);
    digitalWrite(stepM1Pin, LOW);
    delayMicroseconds(80);
  }
```

→ Con il primo if si dice che quando si preme il bottone i motori devono muoversi anche in avanti.

```
for(int x = 0; x < 3200; x++) {
  digitalWrite(stepM2Pin, HIGH);
  delayMicroseconds(80);
  digitalWrite(stepM2Pin, LOW);
  delayMicroseconds(80);
}
```

→ In questo for si fa girare il motore per un giro completo con 3200 impulsi.

Con questi codici si può creare il progetto.

Programma per JJRobots

Le classi utilizzate per il progetto le abbiamo scaricate dal sito di JJRobots e sono le seguenti:

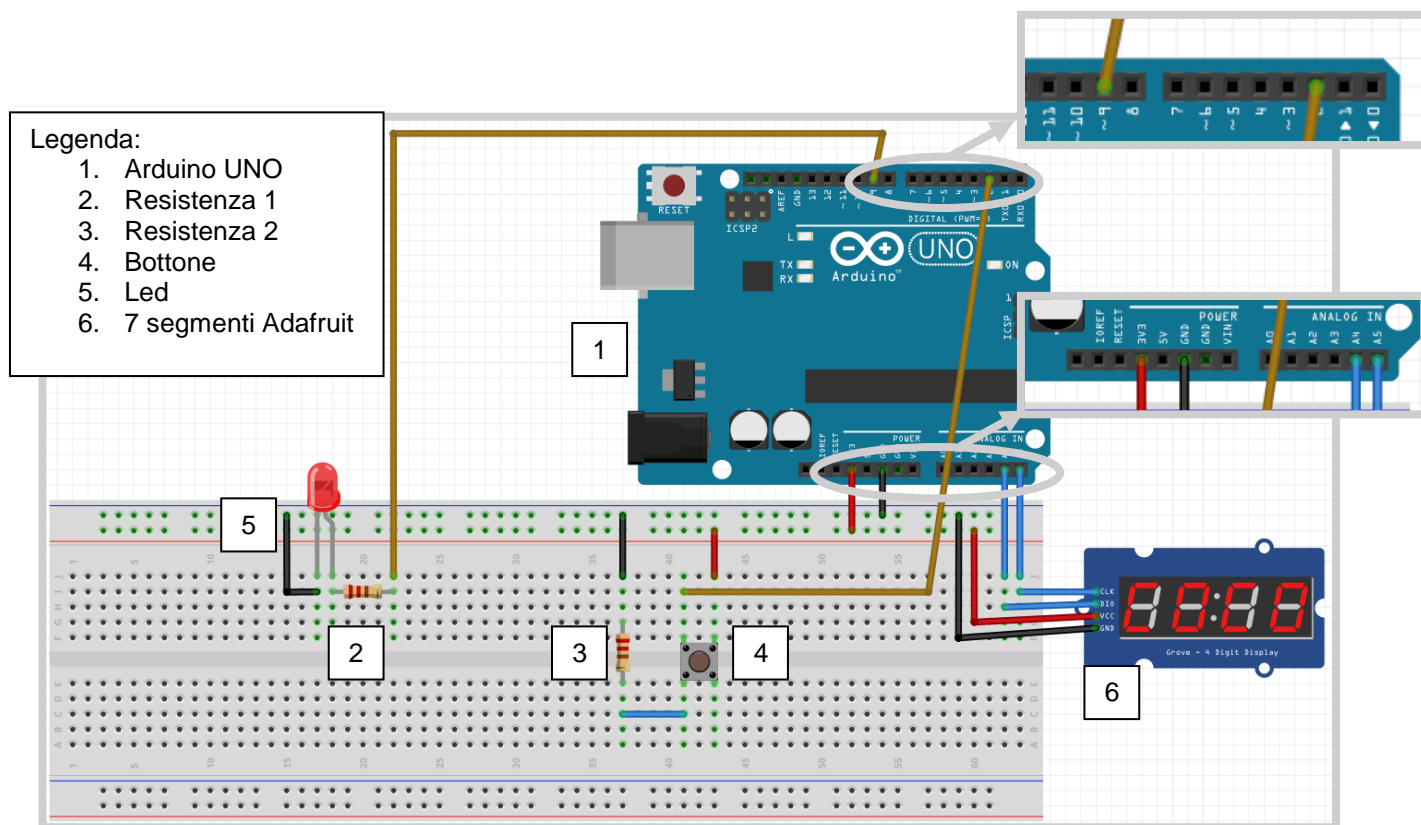


Algoritmo di Timer

Componenti utilizzati

- 1 - Arduino UNO
- 1 – Bottone
- 1 – Led blu
- 2 – Resistenze
- 1 - 7 segmenti Adafruit

Schema di Montaggio



Codice

Variabili:

- | | |
|---------------------------------------------------|----------------------------------|
| • Adafruit_7segment matrix = Adafruit_7segment(); | → 7 segmenti Adafruit |
| • const int MIN_SEC_MINUTE = 0; | → Minimo di secondi e minuti |
| • int mIniziale = 2; | → Durata della partita in minuti |
| • int s1 = MIN_SEC_MINUTE; | → Decine dei secondi |
| • int s2 = MIN_SEC_MINUTE; | → Unità dei secondi |
| • int m = mIniziale; | → Minuti della partita restanti |
| • int t = 60 * m; | → Secondi della partita restanti |

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include "Adafruit_LEDBackpack.h"

Adafruit_7segment matrix = Adafruit_7segment();
```

Importo i pacchetti di Adafruit così da permettere il funzionamento del componente 7 segmenti Adafruit.

```
//Parte al click del bottone
if(digitalRead(buttonPin)){
  int t = 60*m;
  analogWrite(ledPin, LOW);
  for(int i = 0; i < t; i++){
    //Scrive sul 7segmenti
    matrix.drawColon(true);
    matrix.writeDigitNum(0, 0, false);
    matrix.writeDigitNum(1, m, false);
    matrix.writeDigitNum(3, s1, false);
    matrix.writeDigitNum(4, s2, false);
    matrix.writeDisplay();
  }
}
```

Quando il bottone viene schiacciato parte il conto alla rovescia.

Spengo il led che segna la fine della partita.

Il for si ripete per il numero di secondi necessari.

`matrix.drawColon(bool);`

Definisce se i due punti “ : ” devono venire segnati o meno.

`matrix.writeDigit(colonna, numero 0 – 9, bool);`

Colonna: in quale casella devo scriverci su.

Numero: valore da scriverci.

Bool: se il punto del rispettivo numero è acceso o spento.

```
//Conto alla rovescia
if(s1 == MIN_SEC_MINUTE && s2 == MIN_SEC_MINUTE){
  m--;
  s1 = 5;
  s2 = 9;
}else if(s2 == MIN_SEC_MINUTE){
  s1--;
  s2 = 9;
}else{
  s2--;
}

delay(1000);
if(m == MIN_SEC_MINUTE &&
  s1 == MIN_SEC_MINUTE &&
  s2 == MIN_SEC_MINUTE){
  Serial.println("STOP");
  analogWrite(ledPin, HIGH);
}
```

Faccio il conto alla rovescia sottraendo i vari valori; quando sono tutti a 0 si ferma il tempo, quando i secondi (s1, s2) vanno a 0 si sottrae un minuto (m) al tempo totale, se le unità (s2) vanno a 0 si sottrae uno alle decine (s1) e si fa tornare le unità a 9, altrimenti si sottrae uno alle unità (s2).

Delay(1000) serve per far passare un secondo tra una sottrazione e l'altra.

Finito il delay controllo se il tempo è finito e se lo è accendo il led per rendere più visibile la fine.

Montaggio Tavolo

La parte che permette a tutto il progetto di essere operativo e assemblato è il tavolo da gioco, costruito appositamente per l'air ice hockey, fornito di ventola per l'aria e di linee per il gioco.

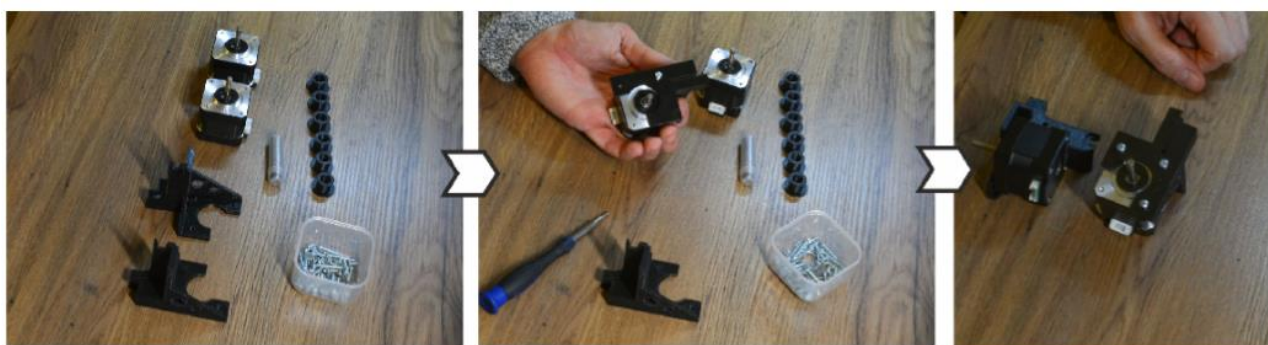
Un tavolo a cui mancano diversi componenti come i motori per permetterne il funzionamento.

I pezzi utilizzati sono stati i seguenti:



Il primo passo da effettuare è l'assemblaggio dei motori.

➔ Materiale necessario: 6x 6mm M3 bolts.

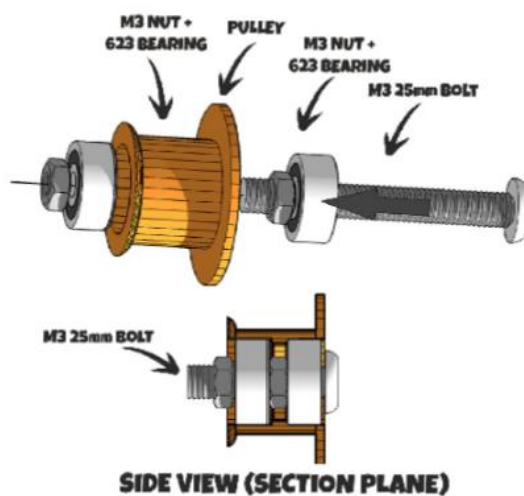


Tramite delle viti fissare i motori ai pezzi di sostegno in plastica.

Successivamente, vanno preparate le pulegge:

➔ Materiale necessario: 12x cuscinetti a sfera 623, 6x 25mm bolt, 6x pulegge.

L'assemblaggio delle sei pulegge va fatto seguendo la seguente illustrazione:



Il passo successivo da fare riguarda l'assemblaggio della sbarra che sostiene il nemico del gioco:

- ➔ Materiale necessario: due sbarre in alluminio, due pezzi in plastica che permettono lo scorrimento del nemico sulle sbarre, due cuscinetti per lo scorrimento di tutto l'assemblaggio perpendicolare alla visione del giocatore che verranno messi all'interno di due pezzi in plastica in cui verranno infilate le sbarre, il pezzo in plastica per il nemico.



Per fare bene questa parte è molto importante fissare con attenzione il pezzo del nemico alle sbarre:

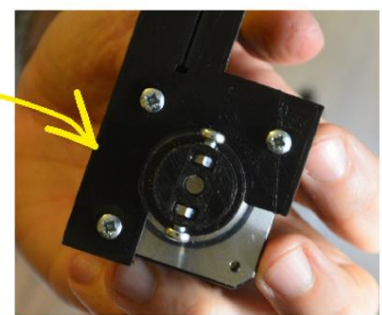
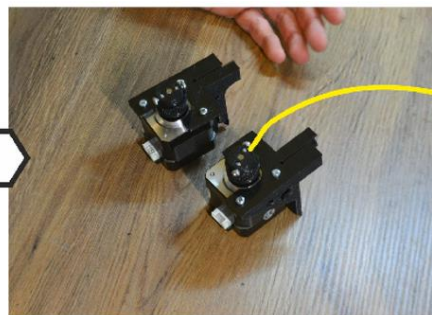


Sotto al pezzo del nemico, va incollata una superficie colorata che ne permette il riconoscimento da parte dell'Arduino.



Adesso devono essere preparate le pulegge da attaccare ai motori:

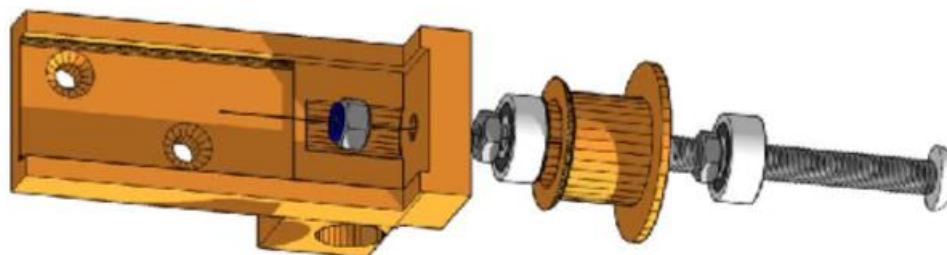
➔ Materiale necessario: 4x 15mm bolts, pezzi che permetteranno alla cinghia di girare.



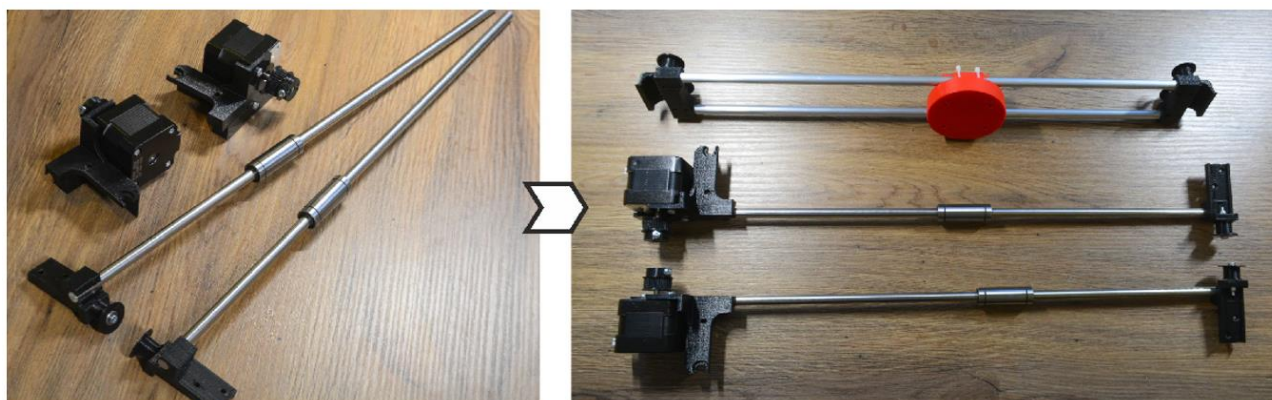
Passiamo a ciò che andrà fissato ai lati del tavolo per far muovere la sbarra centrale:

➔ Materiale necessario: 4x M3 10mm bolts, 4 nuts.

Seguire il procedimento di montaggio come da illustrazione.



Il prossimo passo è molto semplice, si tratta solamente di inserire le sbarre in alluminio all'interno dei sostenitori che verranno poi fissati ai lati del tavolo, per farlo basta seguire le figure sottostanti.

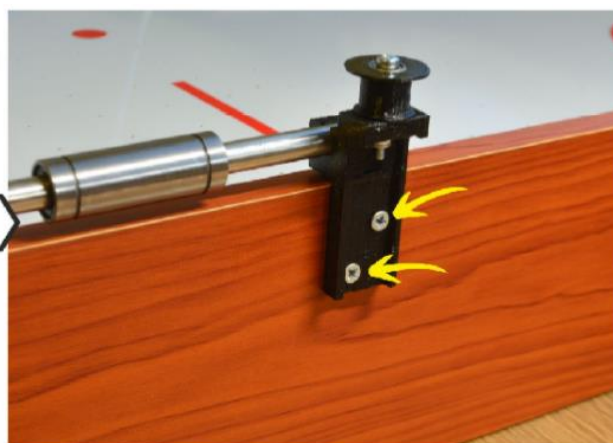


L'ultimo passo per ultimare il lavoro è fissare le sbarre e i motori al tavolo:

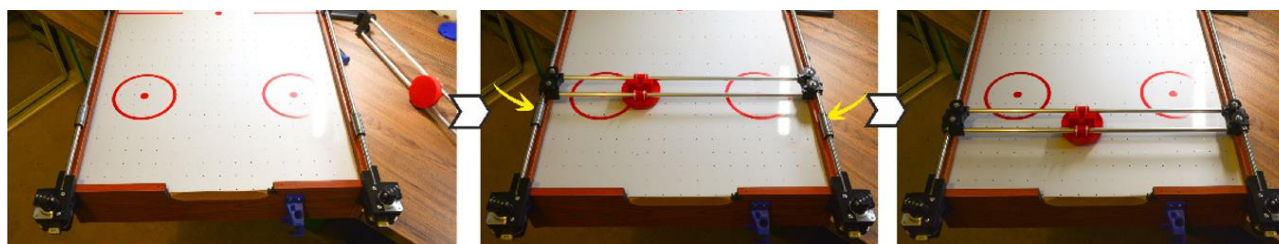


I motori vanno collegati a due angoli del tavolo (angoli del lato dove sta la porta).

Mentre la parte finale della sbarra che parte dal motore va fissata poco prima della metà campo.



Infine attaccare la parte centrale alle laterali, il risultato sarà questo:



3 Test

3.1 Protocollo di test

Test Case:	TC-001	Nome:	Test funzionamento timer
Descrizione:	Quando viene schiacciato il bottone il timer parte e all'arrivo dello 0 si accende il led, per segnalare la fine, e il timer ritorna al valore iniziale.		
Prerequisiti:	Nessuno.		
Procedura:	1. Schiacciare il bottone		
Risultati attesi:	Il timer parte e quando arriva a 0 si accende il led per segnalare che il tempo è finito e ritorna al valore iniziale.		

Test Case:	TC-002	Nome:	Test motori e applicazione JJRobots
Riferimento:	REQ-002		
Descrizione:	Testare il funzionamento dei motori e il collegamento dell'applicazione con Arduino e l'acquisizione dei dati da parte della telecamera.		
Prerequisiti:	Applicazione scaricata su un telefono Android, motori collegati all'Arduino.		
Procedura:	<ol style="list-style-type: none"> 1. Collegare il telefono alla rete Wi-Fi di JJRobots 2. Aprire l'applicazione AirHockey Robot 3. Calibrare i colori e il tavolo 4. Cliccare il bottone start 5. Caricare il codice su Arduino 		
Risultati attesi:	Il robot si muove grazie ai motori in base alla traiettoria del disco		

3.2 Mancanze/limitazioni conosciute

Il progetto non funziona perché ci mancava l'Arduino giusto, noi abbiamo utilizzato il mega quando c'era in realtà bisogno dell'Arduino Leonardo.

La differenza sta nel fatto che l'Arduino prova a leggere i dati dalla scheda dal serial1 ma il serial1 non è collegato alla scheda.

4 Consuntivo

Descrizione	Analisi																							
	1	2	3	10.nov	17.nov	24.nov	1	2	3	01.dic	15.dic	22.dic	1	2	3	12.gen	19.gen							
Raccolta informazioni																								
Analisi Costi + Benefici																								
Analisi Requisiti																								
Progettazione																								
Gantt																								
Implementazione																								
Ricerca di informazioni riguardo l'applicazione JIRobots																								
Studio e montaggio del tavolo e dei pezzi																								
Modifica file read me																								
Raccolgimento dati e motori																								
Algoritmo di punteggio																								
Algoritmo di timer																								
Programma di difesa																								
Programma di attacco																								
Unire attacco e difesa																								
Unire tutti i programmi																								
Test																								
Test algoritmo di punteggio																								
Test algoritmo di timer																								
Test algoritmo di difesa																								
Test algoritmo di attacco																								
Test algoritmo di attacco e difesa																								
Test che tutto funziona																								
Documentazione e presentazione																								
Vengono fatte durante il corso del progetto																								

5 Conclusioni

All'inizio il progetto ci è sembrato molto interessante e fattibile, gli elementi indicati da avere erano chiari, il problema è stato che non ci è stato assegnato l'Arduino Leonardo, quello corretto per poter fare il progetto, quindi non è stato possibile finirlo. Sono state fatte diverse ricerche per fare funzionare l'Arduino Mega al posto di quello Leonardo ma sono state inutili, quindi non funziona nulla eccetto il timer. Eccetto questo il progetto è stato interessante.

5.1 Sviluppi futuri

Non avendo l'Arduino giusto, il prossimo sviluppo prevede di mettere l'Arduino Leonardo e il punteggio verrà fatto tramite dei bottoni messi vicino al giocatore.

5.2 Considerazioni personali

Non avendo i pezzi corretti abbiamo imparato che bisogna seguire ciò che c'è scritto ed insistere nel caso non ci vengano assegnate le cose corrette.

6 Allegati

- Diari di lavoro
- Codici