



**UNIVERSITÀ POLITECNICA DELLE MARCHE**

**FACOLTÀ DI INGEGNERIA**

Corso di Laurea triennale in Ingegneria Informatica e dell'Automazione

**PROGETTAZIONE E SVILUPPO DI ROBOT MARINI PER  
ESPERIENZE DI ROBOTICA EDUCATIVA**

---

**DESIGN AND DEVELOPMENT OF MARINE ROBOTS FOR  
EDUCATIONAL ROBOTICS**

Relatore:

Prof. David Scaradozzi

Tesi di laurea di:

Paradiso Fernando Antonio

**A.A. 2019 / 2020**

# Indice

<b>1. INTRODUZIONE .....</b>	<b>1</b>
1.1 ROBOTICA EDUCATIVA .....	1
1.2 IL PROGETTO ROBOPISCES .....	3
1.3 CONSIDERAZIONI PRELIMINARI.....	4
<b>2. STRUMENTI UTILIZZATI .....</b>	<b>5</b>
2.1 CATIA V5-6 R21 .....	5
2.2 ANALISI FEM.....	7
2.3 ADDITIVE MANUFACTURING .....	8
<b>3. MODELLAZIONE 3D .....</b>	<b>12</b>
3.1 CARATTERIZZAZIONE DEL ROBOT.....	12
3.2 ESPORTAZIONE DEI PROFILI NACA .....	15
3.3 LE PINNE .....	17
3.4 COMPARTI ACCESSORI.....	20
3.5 ASSIEME E DISTINTA BASE.....	25
<b>4. APPLICAZIONE ANALISI FEM .....</b>	<b>29</b>
4.1 ACCORTEZZE NELLA MODELLAZIONE DEI COMPONENTI PER DIMINUIRE LA DEFORMAZIONE.....	29
4.2 PROCEDIMENTO EFFETTUATO.....	30
<b>5. SISTEMA DI NAVIGAZIONE E GUIDA .....</b>	<b>34</b>
5.1 M5STICKC .....	37
5.2 UIFLOW .....	39
5.3 PRIMA ESPERIENZA CON PUPPYC .....	42
5.4 CONTROLLO REMOTO DI PUPPYC ATTRAVERSO L'ESPNOW .....	44
5.5 ADATTAMENTO DELLO SCHEMA DI NAVIGAZIONE E GUIDA A BORRACCIOLO .....	50
<b>CONCLUSIONI.....</b>	<b>54</b>
<b>BIBLIOGRAFIA.....</b>	<b>55</b>
<b>SITOGRAFIA .....</b>	<b>55</b>
<b>RINGRAZIAMENTI.....</b>	<b>56</b>

# 1. INTRODUZIONE

Nei seguenti capitoli illustrerò l'idea di fondo sulla quale si basa la presente tesi. L'uso della tecnologia fin dall'infanzia, se utilizzata in modo corretto e appropriato, può apportare notevoli vantaggi dal punto di vista formativo di un bambino. Secondo alcuni articoli infatti, i bambini con autismo ad esempio, hanno una naturale propensione ad interagire con i robot che possono essere usati proprio come uno strumento riabilitativo che faccia da tramite tra il terapeuta e il bambino. È secondo queste esigenze che nasce il progetto Educational Robotics e il concetto di robotica educativa.

## 1.1 ROBOTICA EDUCATIVA

Le competenze digitali stanno diventando sempre più centrali nella politica educativa di ogni paese. I sistemi educativi stanno affrontando nuove sfide nell'era della 4<sup>a</sup> rivoluzione industriale. Tutti gli alunni dovrebbero acquisire conoscenze, abilità e competenze per entrare con successo nel mercato del lavoro e avere un ruolo attivo nella società. Competenze chiave come l'alfabetizzazione, la matematica e le conoscenze scientifiche di base sono abilità essenziali per una vita professionale e civile di successo. Nell'ultimo decennio molti progetti hanno dimostrato come l'Educational Robotics (ER) può essere un potente strumento che promuove e sviluppa il pensiero creativo, il lavoro di squadra, le capacità di risoluzione dei problemi e la motivazione dei bambini. L'attività di programmazione e costruzione di robot è un modo ideale per introdurre tecnologia e capacità ingegneristiche: può aiutare ad esempio i bambini a sviluppare il pensiero computazionale o la capacità di elaborare prodotti e trovare delle soluzioni. Tuttavia sono necessarie ulteriori ricerche per valutare l'uso della robotica come mezzo per lo sviluppo delle competenze o per creare strumenti di valutazione facilmente riutilizzabili dagli insegnanti per le loro attività, convalidando così l'impatto della robotica nella promozione dell'apprendimento e dello sviluppo delle competenze degli studenti.

Molti ricercatori hanno studiato l'effetto che determina l'utilizzo dei robot come supporto all'istruzione: gli studi hanno dimostrato che i robot possono aiutare gli studenti a sviluppare capacità di risoluzione dei problemi e apprendere la programmazione informatica, la matematica e la scienza. L'approccio educativo basato principalmente sullo sviluppo della logica e della creatività nelle nuove generazioni è molto promettente sin dalla prima fase dell'educazione; in tal proposito l'uso dei sistemi robotici sta diventando fondamentale se

questi vengono applicati sin dalla prima infanzia. Nelle scuole primarie la programmazione dei robot è divertente e rappresenta quindi uno strumento eccellente sia per introdurre le TIC (tecnologie dell'informazione e della comunicazione) sia per aiutare lo sviluppo delle capacità logiche e linguistiche dei bambini. Negli ultimi cinque anni il Ministero della Pubblica Istruzione italiano ha focalizzato la sua attenzione sulle competenze digitali, riconosciute come fondamentali e indispensabili per la crescita dei futuri cittadini nell'era dell'informazione, sostenendo quindi progetti volti a sviluppare il pensiero computazionale e la creatività digitale nelle scuole.

## 1.2 IL PROGETTO ROBOPISCES

Proporre attività di robotica marina alla scuola primaria e secondaria di primo grado è molto importante per facilitare lo sviluppo di una coscienza ambientale negli studenti, che grazie alla robotica educativa progettano e costruiscono prototipi per risolvere problemi attuali e futuri (inquinamento, protezione della biodiversità, etc.).

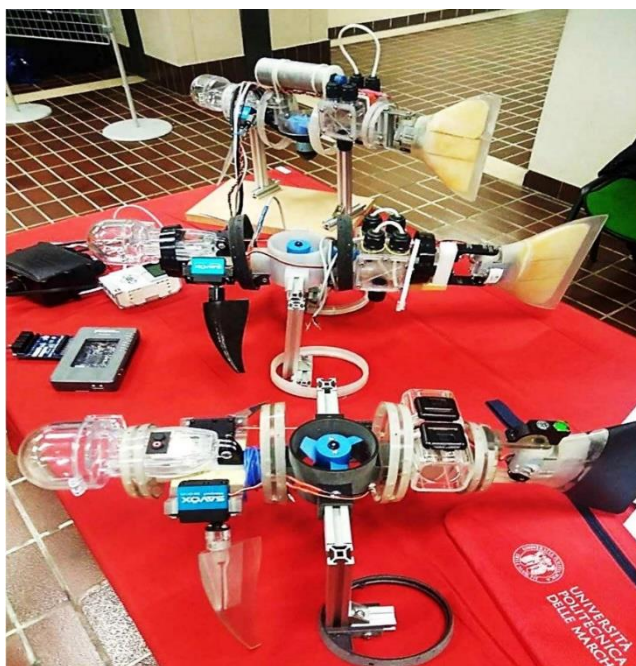
Il progetto RoboPisces è un progetto europeo frutto di una collaborazione tra istituzioni accademiche e scuole di istruzione primaria che, attraverso la robotica educativa, porta competenze digitali e conoscenza delle materie STEAM (Scienza, Tecnologia, Ingegneria, Arte e Matematica) nei banchi di scuola, al fine di avvicinare gli studenti al campo della robotica.

L'obiettivo è quello di portare gli studenti a saper costruire e programmare robot marini didattici tramite hardware e software dedicati di semplice utilizzo a partire dalle unità LEGO Mindstorms EV3, estendendo inoltre il lavoro alla programmazione su scheda Arduino. In questo modo si incoraggiano gli studenti ad usare la loro immaginazione e a giovare delle competenze che vengono acquisite man mano durante il corso di robotica educativa.

Nell'ambito di questa tesi, verrà preso in esame il lavoro da me svolto nel periodo di tirocinio, il quale ha compreso la progettazione e costruzione di un nuovo prototipo di robot pesciforme rivolto al progetto "Robopisces", utilizzando come corpo del pesce una borraccia UNIVPM.



*Figura 1.2.1 - Un bambino programma con Lego Mindstorm*



*Figura 1.1.2 - Progetto Robopisces*

### 1.3 CONSIDERAZIONI PRELIMINARI

I progetti Borracciolo, nascono dall'idea di modificare un precedente prototipo di Guizzo 5.0 sostituendo il corpo con la parte centrale di una borraccia UNIVPM che andrà ad ospitare la componente circuitale e la batteria.

È stato quindi ripreso il modello CAD della versione 5.0 e tenendo conto degli sviluppi delle versioni successive, si è cercato di unificare il più possibile i vari componenti pur mantenendo una certa versatilità per quanto riguarda l'inserimento o l'eventuale sostituzione di qualche pezzo. La maggior parte delle componenti del robot sono state pensate per essere costruite con la stampante 3D STRATASYS in dotazione al dipartimento dell'ingegneria dell'informazione dell'Università Politecnica delle Marche, utilizzando materiali compositi in grado di fotopolimerizzare, in modo da ridurre anche i costi di materie prime ordinate da terzi. Oggetto di questa tesi, sarà controllare le varie pinne che compongono il robot in modo da simulare il movimento di un pesce in acqua. Inoltre la modellazione delle varie parti è stata pensata con lo scopo di:

- Rispettare il più possibile quello che è il profilo NACA (cap. 4.2);
- Eliminare tutti gli spigoli vivi presenti in alcuni componenti del Guizzo 5.0;
- Bilanciare il più possibile i vari pezzi del pesce, alleggerendo alcuni componenti per favorire il galleggiamento;
- Raggiungere considerevoli profondità garantendo la massima impermeabilità.



*Figura 1.3.1 - Borraccia UNIVPM usata per il corpo del prototipo*



*Figura 1.3.1 - Guizzo 5.0*

## **2. STRUMENTI UTILIZZATI**

Verranno qui descritti gli strumenti digitali e di laboratorio che sono stati utilizzati per costruire i due robot. Piattaforma fondamentale usata per la creazione del prototipo è stata Catia, la quale permette oltre che progettare il modello e di avere quindi una buona visione virtuale del prodotto che si andrà a creare, anche di effettuare un'analisi FEM dei vari componenti. Quest'ultimi sono stati stampati con una stampante 3D Stratasys, che sfrutta il processo di produzione additiva.

### **2.1 CATIA V5-6 R21**

Il disegno del progetto su computer è di fondamentale importanza per quanto riguarda la progettazione di un prototipo, in quanto permette di poter ridurre drasticamente i tempi e i costi di sviluppo mantenendo comunque il comportamento dei modelli creati molto vicino a quello di un solido reale: nell'ambito CAD (Computer Aided Design) infatti l'oggetto fisico che si vuole creare o che già esiste viene rappresentato senza ambiguità nelle sue forme e nelle sue dimensioni, permettendo per di più di archiviare e utilizzare precedenti modelli, già dotati di quote tecniche e calcoli statici o dinamici, su cui basare un nuovo progetto.

Esistono vari sistemi CAD con i quali è possibile creare il modello e modificarlo molteplici volte, eseguire verifiche, analisi strutturali (FEM/FEA) e conversione del file in formato STL (Standard Triangulation Language) per la stampa del modello stesso, ovvero la stereolitografia che consiste nel realizzare singoli oggetti tridimensionali a partire direttamente da dati digitali elaborati dal software CAD/CAM. Questi non sono solo in grado di supportare operazioni relative ad aspetti geometrici e dimensionali del progetto ma, grazie all'analisi di modelli matematici, permettono di effettuare un'analisi immediata di proprietà fisiche come la massa, il volume, la densità, il baricentro e i momenti principali di inerzia, definendo anche il tipo di materiale che si va ad utilizzare per ogni parte del modello. Come anticipato, il software utilizzato per questo progetto è stato CATIA V5-6 R21 (Computer Aided Three dimensional Interactive Application), sviluppato nel 1977 dall'azienda francese Dassault Systèmes per uso interno nell'ambito della produzione di aerei, e reso pubblico dal 1998, tanto da essere il software più usato nell'industria tecnica ingegneristica, specialmente nei settori automobilistici, marini e aerospaziali.

Il software è stato scelto per le caratteristiche che possiede. Infatti CATIA è un sistema altamente scalabile, pertanto può essere corredato di altre applicazioni o moduli comprendenti tutti gli strumenti normalmente disponibili per questa fascia di sistemi PLM come, ad esempio, piattaforme PDM (tra le quali si ricordano Enovia e Smarteam), ambienti CAE per l'analisi agli elementi finiti, ecc. Inoltre è un modellatore ibrido, ovvero la rappresentazione solida e quella a superfici possono convivere, dando vita ad un unico modello. Ciò semplifica le operazioni necessarie in varie attività come ad esempio quella di ingegnerizzazione delle forme.

Gli ambienti che sono stati usati principalmente in questo progetto sono stati:

- Part design;
- Assembly design;
- Drafting;
- Generative structural analysis.

L'ambiente Part design si caratterizza principalmente per la creazione 2d di uno schizzo su uno dei tre piani lungo le direzioni x, y, z e per la trasformazione di quest'ultimo in un prisma. Con le varie altre funzioni è possibile creare un pezzo in modo preciso e impeccabile, salvo poi le imprecisioni che ci saranno durante la stampa 3d.

L'ambiente Assembly design è molto importante in quanto vengono assemblati tutti i pezzi che sono stati modellati separatamente, in quello che sarà poi il robot finale che vogliamo creare. Inoltre alcuni dei componenti, sono stati creati direttamente nell'ambiente assembly grazie alla possibilità di usare contemporaneamente il part design per creare così un pezzo in un tempo minore, dato che ci si rende conto del risultato man mano che lo si costruisce. Quindi cosa molto importante è proprio la trasversalità di questi ambienti durante il processo di modellazione.

L'ambiente Drafting, permette di mettere in tavola i componenti in modo chiaro e con l'ausilio delle quote si possono leggere facilmente tutte le varie misure che possono essere utili a chi deve prendere in mano un progetto per modificarlo o per migliorarlo.

L'ambiente Generative structural analysis permette di effettuare un'analisi FEM la cui descrizione generale è illustrata nel prossimo capitolo.

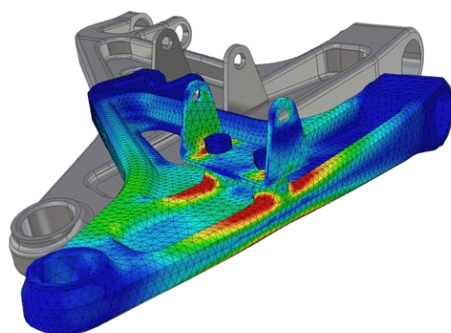


*Figura 2.1 - Logo Catia*



## 2.2 ANALISI FEM

Il metodo degli elementi finiti FEM (Finite Element Method) è una tecnica numerica atta a cercare soluzioni approssimate di problemi descritti da equazioni differenziali alle derivate parziali riducendo queste ultime a un sistema di equazioni algebriche. In generale, il metodo agli elementi finiti si presta molto bene a risolvere equazioni alle derivate parziali quando il dominio ha forma complessa (come il telaio di un'automobile o il motore di un aereo), quando il dominio è variabile (per esempio una reazione a stato solido con condizioni al contorno variabili), quando l'accuratezza richiesta alla soluzione non è omogenea sul dominio (in un crash test su un autoveicolo, l'accuratezza richiesta è maggiore in prossimità della zona di impatto) e quando la soluzione cercata manca di regolarità. Inoltre, il metodo è alla base dell'analisi agli elementi finiti.

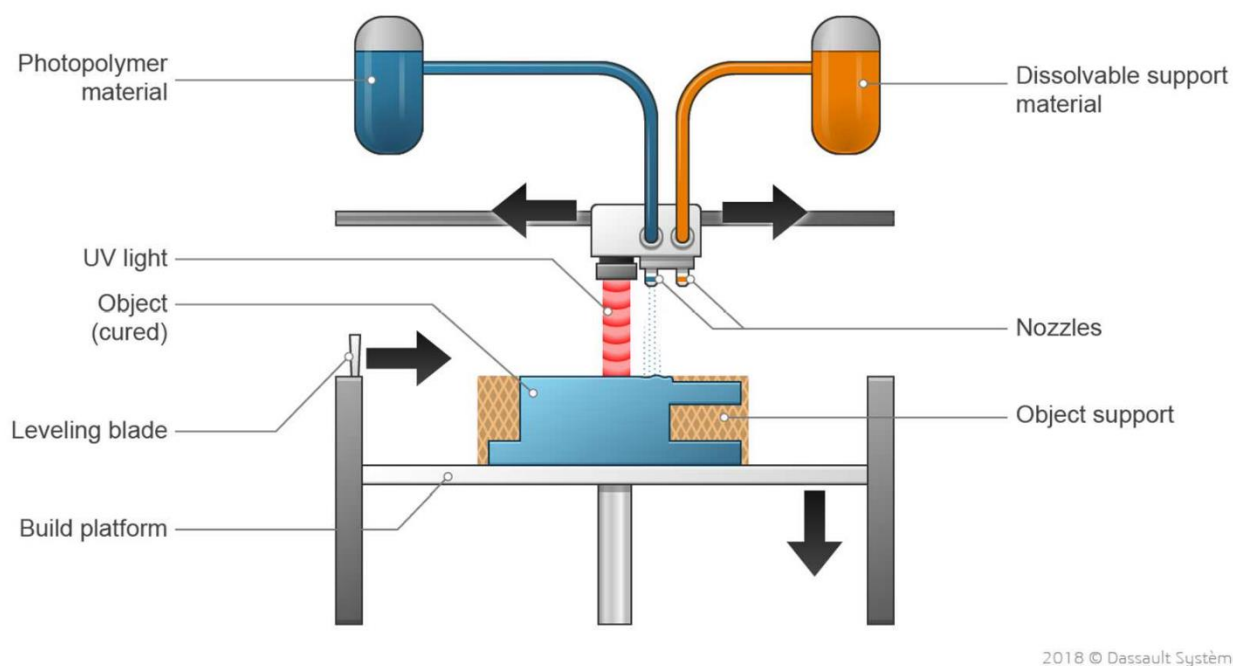


*Figura 2.2 - Logo analisi fem Catia*

Il programma di simulazione usato nel mio caso è stato CATIA V5-6. Tenendo conto di azioni meccaniche dovute a forze ed accelerazioni nell'acqua, è stato possibile modificare nel modo più consono alcuni pezzi che necessitavano di una maggiore resistenza, attraverso l'inserimento di nervature o inspessimenti. Quindi con il software in questione è stato possibile analizzare i carichi sul contro coperchio verificando la deformazione. Nel capitolo 4 verranno mostrati più nel dettaglio i vari passaggi e la soluzione adottata.

## 2.3 ADDITIVE MANUFACTURING

Come accennato in precedenza, la maggior parte delle parti del robot è stata creata su CAD per poter essere realizzata mediante tecniche di produzione additiva o produzione a strati usando un fotopolimero che solidifica all'esposizione dei raggi UV (figura 4).



*Figura 2.3.1 - Additive Manufacturing with UV light*

Il processo di produzione additiva ha come input la realizzazione del modello 3D dell'oggetto (progettazione CAD). Viene quindi svolto un processo semi-automatico (svolto oramai da tutti i più diffusi software di progettazione) di conversione del file in formato STL (Standard Triangulation Language), standard grafico che descrive la superficie dell'oggetto mediante decomposizione in triangoli il cui numero è tanto maggiore quanto meglio si vuole approssimare la superficie.

Successivamente, attraverso il software di modellazione si procede alla scomposizione dell'oggetto in strati (layer) stampabili dalle stampanti 3D. La stampante in questione utilizzata è stata la STRATASYS OBJET30 PRIME con tecnica di fotopolimerizzazione. Il funzionamento è il seguente:

## 1. Preparazione della stampa:

1.1. Si carica il file STL all'interno del software di stampa e si definiscono le impostazioni del modello e del vassoio. È possibile realizzare l'oggetto con una finitura ruvida (matte) o lucida (glossy) che è quella che caratterizzerà tutti i pezzi del robot per caratteristiche estetiche e funzionali (infatti l'acqua scivola meglio su una superficie liscia anziché ruvida). Per quanto riguarda le impostazioni del vassoio di stampa invece, si suddividono in:

-Posizionamento: durante questa fase, si sceglie con cura la posizione e l'orientamento del modello 3D. Questo perché un oggetto prevalentemente verticale implica un dispendio maggiore di resina che verrà successivamente persa, mentre se posizionato orizzontalmente, c'è un guadagno in termini di tempo e di supporto utilizzato in quanto i layer sono minori;

-Validazione: il software effettua un controllo sulla posizione degli oggetti in modo tale che non si sovrappongano o non ci siano problemi di modellazione;

-Stima: viene effettuata in questo caso una stima abbastanza fedele (dopo varie verifiche) dei costi e dei tempi impiegati per stampare il tutto. Anche in questo caso bisogna fare attenzione al valore che ci riporta il software in quanto il materiale di supporto ha un costo maggiore della resina vera e propria che compone l'oggetto (Veroblack plus) e nel caso ci siano eccessi, bisognerà ripetere con più attenzione la fase di posizionamento;

-Esecuzione: si avvia la stampa.

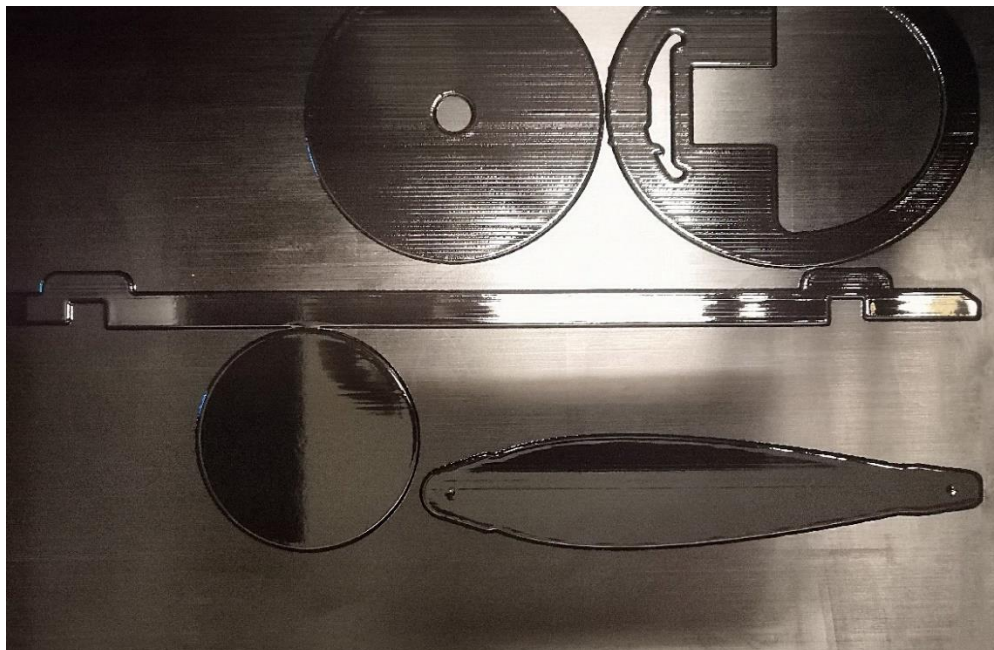
1.2. Il software converte l'STL scomponendolo in strati (layer) di stampa; una definizione maggiore dell'oggetto si può avere con un numero più elevato di strati ma questo comporta un dispendio maggiore di tempo, materiale ed energie.

1.3. La stampante viene caricata con due tipi di liquidi. Il fotopolimero attivato da una luce UV e un secondo materiale, solubile, utilizzato per gli elementi di supporto a perdere.

1.4. La testina di stampa viene riscaldata e ripulita dagli scarti precedenti, l'operazione può durare dai 10 ai 25 minuti se la stampante è stata spenta per molto tempo.

## 2. Fase di stampa

- 2.1. La testina si sposta sul piano di lavoro depositando un primo strato di resina di supporto a perdere e poi inizia con i layer (Figura 3.3.2).



*Figura 2.3.2 - Particolare del piano di stampa con i primi strati di resina di supporto*

- 2.2. Nella creazione dell'oggetto il liquido viene depositato e mediante l'induzione di energia emessa dalla luce UV, i monomeri si assemblano in catene più lunghe generando il polimero.
- 2.3. Durante questa fase quindi, qualsiasi foro, sezione cava o parete dell'oggetto viene rivestita dalla resina di supporto per impedire il collasso del modello creato.

## 3. Fase di pulizia e finitura

- 3.1. Una volta terminato il processo di stampa, a seconda della finalità d'uso, sono necessarie attività di post-produzione e finitura per completare il manufatto realizzato. Il pezzo viene staccato dal piano di lavoro e con l'aiuto di una macchina idropulitrice viene eliminato il materiale di supporto. Per completare bene questa fase, viene immerso ulteriormente per circa 4 ore in una soluzione alcalina in grado di sciogliere il restante supporto difficile da togliere manualmente. Infine si risciacqua ancora con l'idropulitrice curando i dettagli dell'oggetto (come ad esempio i vari fori non completamente liberi).

3.2. Il piano di stampa viene ripulito dagli scarti con dell'alcool isopropilico e preparato per un nuovo processo di stampa.

La Stratasys offre diversi materiali per la stampa di oggetti. Il progetto di Borracciolo 1.0 e 2.0 è stato realizzato con parti in VEROBLOCK, un fotopolimero PolyJet nero che è un simil-polipropilene caratterizzato da bassa elasticità e da allungamento a rottura del 10÷25%. Offre robustezza, rigidità e resistenza agli urti nonché la capacità di miscelarsi ad altri materiali per una varietà di opacità, tonalità e durezza.

Altro materiale adoperato per i pezzi che non sono stati modificati, è il Veroclear, un fotopolimero PolyJet trasparente che simula il PMMA (polimetilmetacrilato), più comunemente noto come acrilico chiaro. Esso consente di vedere componenti e caratteristiche interne di un oggetto diventando quindi un buon sostituto del vetro grazie alle sue caratteristiche estetiche che lo rendono trasparente.

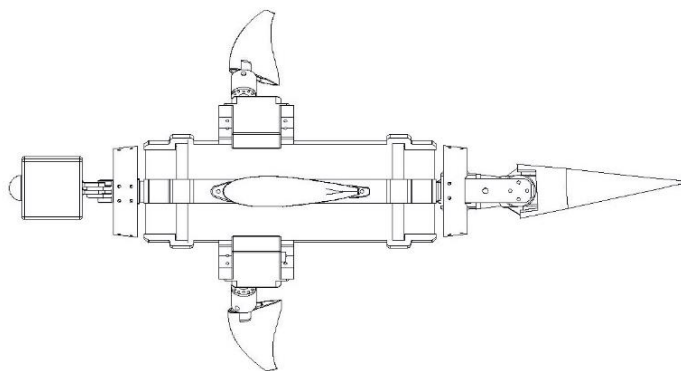
Il materiale di supporto solubile usato invece è il SUP706, studiato specificamente per agevolare la pulizia di piccole parti, con lo scopo di ridurre il tempo e il lavoro necessario per la pulizia dei pezzi stampati.

### 3. MODELLAZIONE 3D

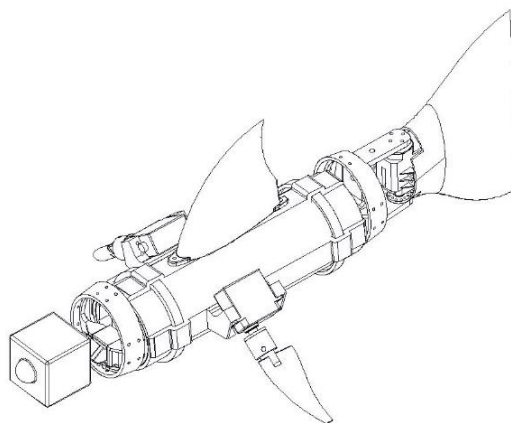
In questo capitolo parlerò di come è costituito il prototipo presentandolo da più punti di vista attraverso le bozze create su Catia, definendo quali sono stati i pezzi usati e come sono stati creati. Infatti le pinne ma anche l'intera struttura che compongono il robot, sono il frutto dell'estrazione del profilo NACA, che è un particolare profilo alare di ali di aeromobili studiati dalla National Advisory Committee for Aeronautics statunitense. Le sue caratteristiche sono descritte dai valori che compaiono nel nome, che se inseriti in opportune equazioni permettono di calcolare tutte le proprietà della sezione alare (curvatura, ventre, ecc). Verrà poi illustrata la morfologia di un pesce descrivendo le pinne modellate il più simile possibile a quelle reali e realizzate in modo da essere perfettamente compatibili con i pezzi del prototipo.

#### 3.1 CARATTERIZZAZIONE DEL ROBOT

I modelli seguono in maniera approssimata la forma di un pesce e sono quindi caratterizzati dal corpo centrale di una borraccia e dalle principali pinne che ne caratterizzano il movimento in acqua. Borracciolo 1.0 presenta un corpo con la borraccia tagliata a entrambi i lati, mentre Borracciolo 2.0 mantiene la parte posteriore della borraccia completa.



*Figura 3.1.1 - Borracciolo 1.0, vista dall'alto*



*Figura 3.1.2 - Borracciolo 1.0, vista isometrica*

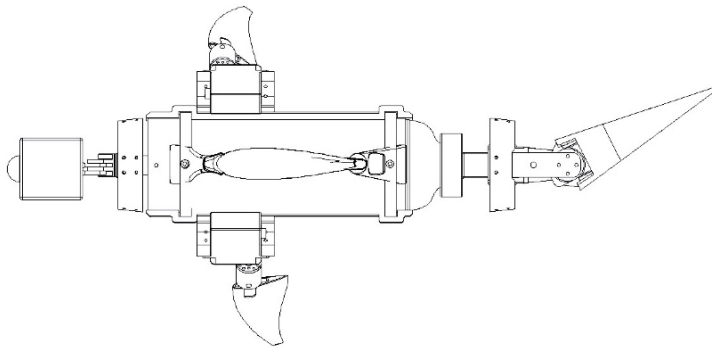


Figura 3.1.3 - Borracciolo 2.0, vista piano superiore

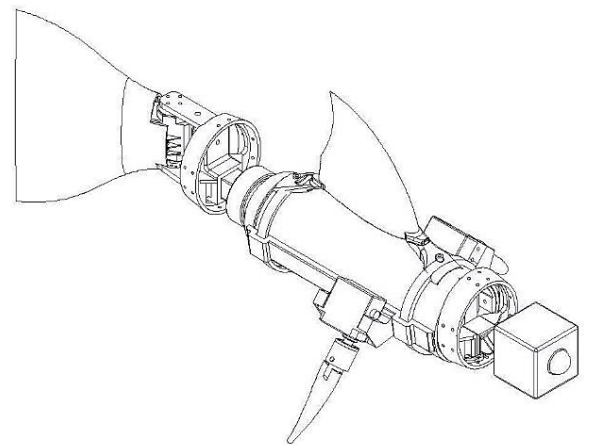


Figura 3.1.4 - Borracciolo 2.0, vista isometrica

Di seguito si porta in esame il processo di trasformazione eseguito partendo dai dati CAD del modello Guizzo 5.0 della figura 3.1.5 arrivando alle versioni di Borracciolo 1.0 e 2.0 della figura 3.1.6 e 3.1.7.

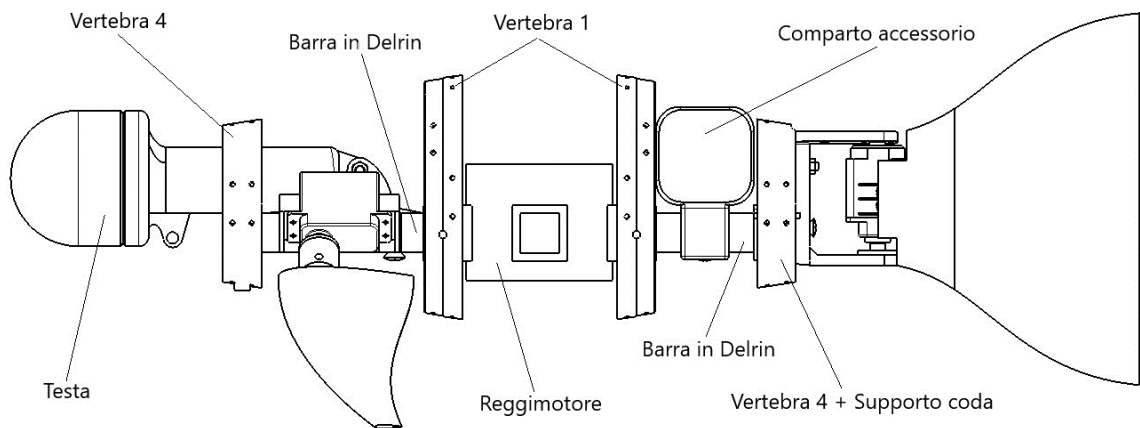


Figura 3.1.5 - Guizzo 5.0

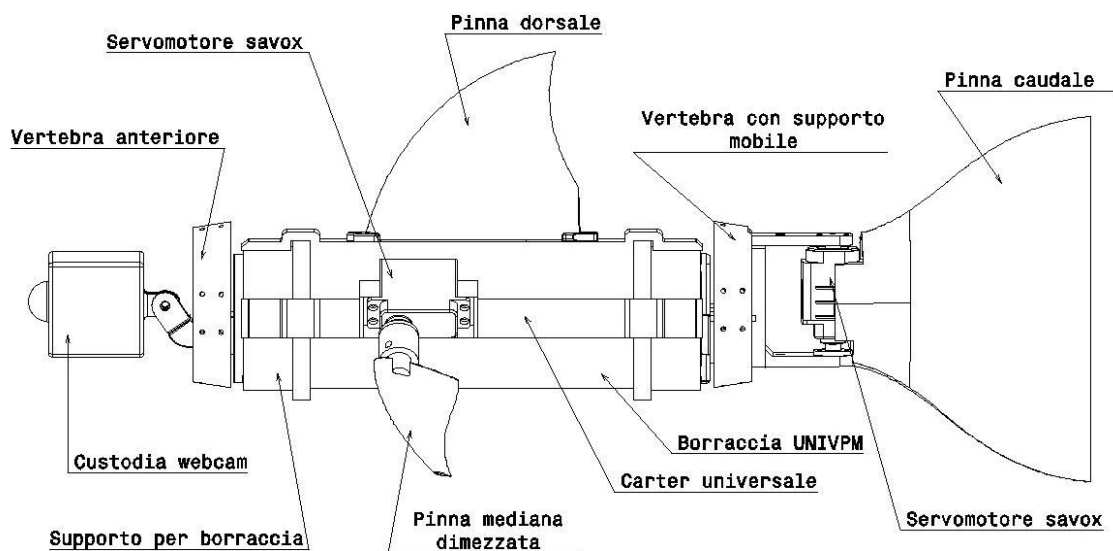


Figura 3.1.6 - Borracciolo 1.0 con particolari

Dalle immagini appena mostrate, è possibile notare come il nuovo modello sia più corto in termini di lunghezza, che passa da 585.1mm a 558.0mm. La sezione anteriore è ora un tutt'uno con il corpo, in quanto i supporti per i servomotori sono innestati direttamente alla borraccia osservando una posizione delle pinne mediane più centralizzata (con la possibilità di variarla grazie ai vari fori inseriti nel porta carter universale).

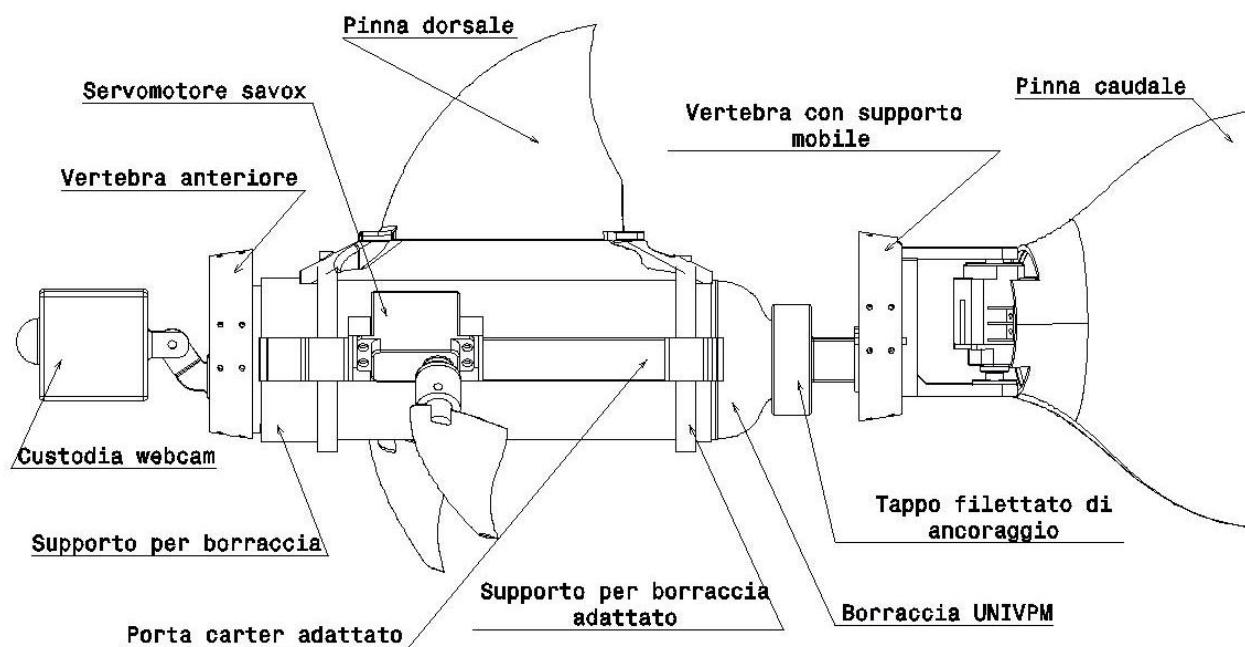


Figura 3.1.7 - Borracciolo 2.0 con particolari

Il secondo prototipo, presenta un corpo di lunghezza pari a 586.9 mm, quindi quasi uguale alla lunghezza di Guizzo 5.0. Per non sbilanciare il nuovo robot in modo eccessivo, si è ritenuto opportuno spostare più in avanti i due servomotore savox con le relative pinne. contiene inoltre un tappo filettato di ancoraggio per la vertebra posteriore.

Le vertebre utilizzate in entrambe le versioni, sono la *vertebra 4* a prua e la sua simmetrica a poppa che ospita la trave di afferraggio superiore atta far ruotare il servomotore in posizione verticale attorno l'asse del vincolo. È stato eliminato il reggimotore e le due vertebre centrali, comportando quindi l'assenza del motore brushless e del movimento verticale; è stata cambiata la pinna dorsale e la custodia per la gopro.



### 3.2 ESPORTAZIONE DEI PROFILI NACA

L'intera struttura delle vertebre del robot è costruita seguendo un solido di rivoluzione fittizio basato su un profilo NACA.

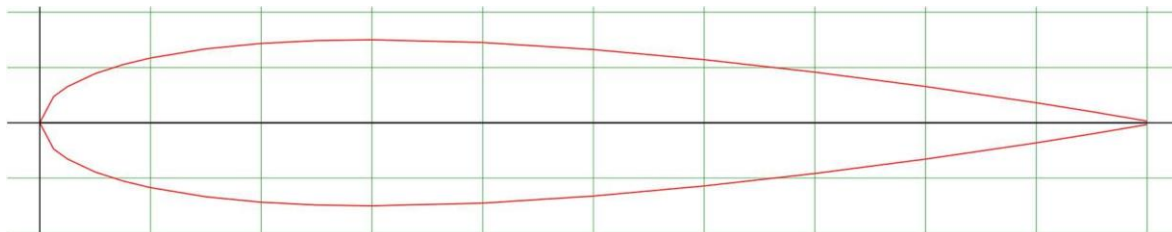


Figura 3.2.1 - Profilo NACA0015

La curva della figura 3.2.1 può essere tradotta (considerando solo una metà del profilo essendo questi simmetrico) in questo insieme di punti lungo i 3 assi principali:

x	y	z
1	0,00158	0
0,95	0,01008	0
0,9	0,0181	0
0,8	0,03279	0
0,7	0,0458	0
0,6	0,05704	0
0,5	0,06617	0
0,4	0,07254	0
0,3	0,07502	0
0,25	0,07427	0
0,2	0,07172	0
0,15	0,06682	0
0,1	0,05853	0
0,075	0,0525	0
0,05	0,04443	0
0,025	0,03268	0
0,0125	0,02367	0
0	0	0

I valori delle coordinate rappresentano un profilo di dimensione unitaria e quindi con punti compresi nell'intervallo  $[0;1]$ . Per importare su CATIA una curva delle dimensioni desiderate, si procederà tramite Excel a moltiplicare ciascun valore per la lunghezza della corda del profilo che si vuole ottenere. Una volta ottenuti i valori, si potrà procedere importando come Macro la tabella di dati costruendo la curva su CATIA. Questo procedimento è universale per tutti i tipi di curve che si vogliono importare come Sketch di CATIA.

Per quanto riguarda il nostro progetto, per estrapolare i modelli delle vertebre da modificare, è stato necessario partire dal file CAD del corpo di Guizzo 2.0, precedente al Guizzo 5.0, visibile nella figura 4.2.2.

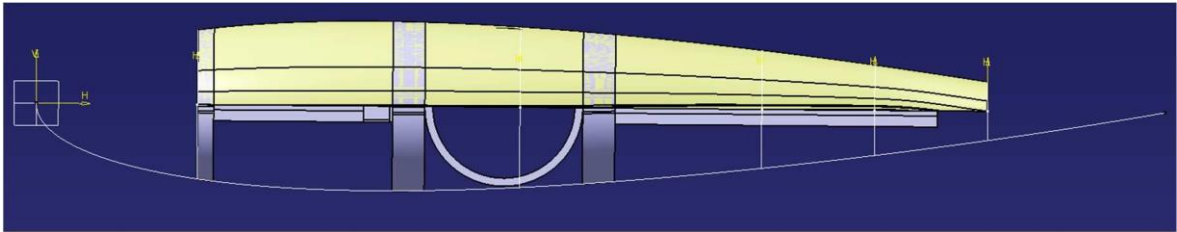


Figura 3.2.2 - Profilo NACA su CATIA con superficie e vertebre, vista dall'alto.

Nell'immagine è ben visibile il profilo NACA importato da Excel, una metà della superficie ottenuta come rivoluzione del profilo (in giallo) e le varie vertebre di colore grigio ottenute a partire dalla superficie.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1								StartLoft						
2	x	y	z					StartCurve						
3	1	0.00158	0		Chord Line	150		150	0.237	0				
4	0.95	0.01008	0					142.5	1.512	0				
5	0.9	0.0181	0					135	2.715	0				
6	0.8	0.03279	0					120	4.9185	0				
7	0.7	0.0458	0					105	6.87	0				
8	0.6	0.05704	0					90	8.556	0				
9	0.5	0.06617	0					75	9.9255	0				
10	0.4	0.07254	0					60	10.881	0				
11	0.3	0.07502	0					45	11.253	0				
12	0.25	0.07427	0					37.5	11.1405	0				
13	0.2	0.07172	0					30	10.758	0				
14	0.15	0.06682	0					22.5	10.023	0				
15	0.1	0.05853	0					15	8.7795	0				
16	0.075	0.0525	0					11.25	7.875	0				
17	0.05	0.04443	0					7.5	6.6645	0				
18	0.025	0.03268	0					3.75	4.902	0				
19	0.0125	0.02367	0					1.875	3.5505	0				
20	0	0	0					0	0	0				
21	0.0125	-0.02367	0					1.875	-3.5505	0				
22	0.025	-0.03268	0					3.75	-4.902	0				
23	0.05	-0.04443	0					7.5	-6.6645	0				
24	0.075	-0.0525	0					11.25	-7.875	0				
25	0.1	-0.05853	0					15	-8.7795	0				
26	0.15	-0.06682	0					22.5	-10.023	0				
27	0.2	-0.07172	0					30	-10.758	0				
28	0.25	-0.07427	0					37.5	-11.1405	0				
29	0.3	-0.07502	0					45	-11.253	0				
30	0.4	-0.07254	0					60	-10.881	0				
31	0.5	-0.06617	0					75	-9.9255	0				
32	0.6	-0.05704	0					90	-8.556	0				
33	0.7	-0.0458	0					105	-6.87	0				
34	0.8	-0.03279	0					120	-4.9185	0				
35	0.9	-0.0181	0					135	-2.715	0				
36	0.95	-0.01008	0					142.5	-1.512	0				
37	1	-0.00158	0					150	-0.237	0				
38								EndCurve						
39								EndLoft						
40								End						
41														

Figura 3.2.3 - Esempio di conversione dei valori unitari con una corda pari a 150

Quindi ricapitolando, il profilo NACA è stato usato all'interno del progetto per:

- La struttura delle vertebre;
- La base delle varie pinne che compongono il prototipo;
- Dare una determinata forma al modello realizzato.

### 3.3 LE PINNE

Le pinne rappresentano i principali organi di equilibrio e locomozione delle specie animali acquatiche, formate da una membrana sostenuta da raggi scheletrici, ossei o cartilaginei. Ogni pinna è responsabile di tutti diversi movimenti dell'animale, solitamente, quasi tutti i pesci, hanno sette pinne, tre singole e due a coppia.

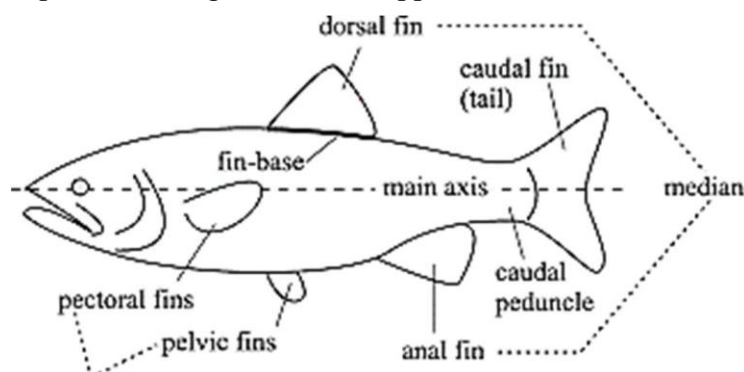
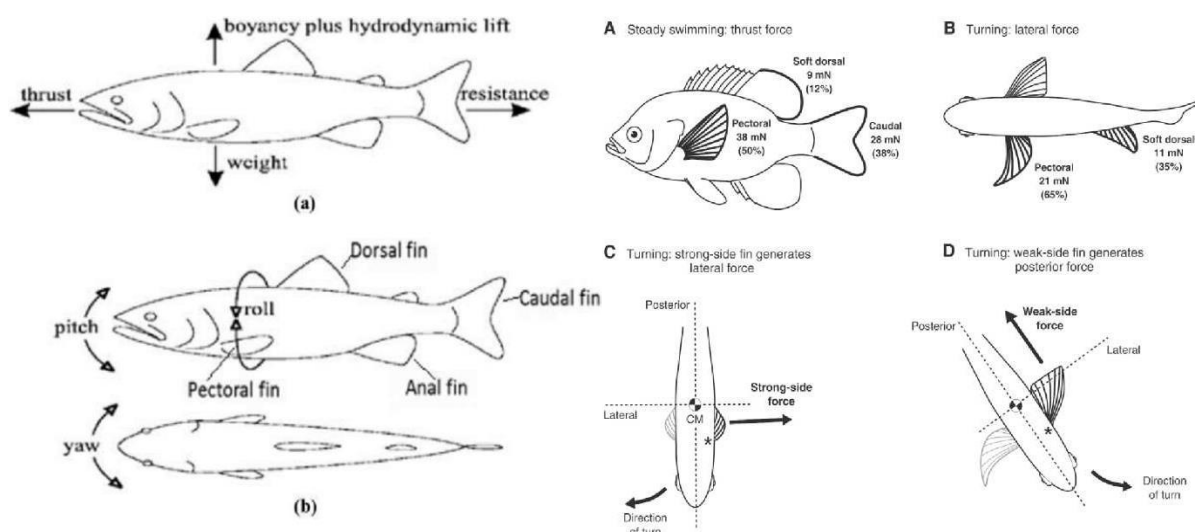


Figura 3.2.1 - Morfologia di un pesce

Tre sono le pinne prese in considerazione per il progetto di Borracciolo:

- Pinne pettorali (o mediane) che situate nella parte inferiore del corpo, generalmente dietro le branchie, servono a far mantenere un assetto lineare all'animale durante la sua nuotata ed entrano anche in gioco quando il pesce deve muoversi con una corretta precisione;
- Pinna caudale, chiamata anche coda del pesce, è senza dubbio quella più importante per la sua locomozione, la quale contribuisce maggiormente alla sua spinta in avanti;
- Pinna dorsale, il cui principale compito è quello di stabilizzare l'animale.



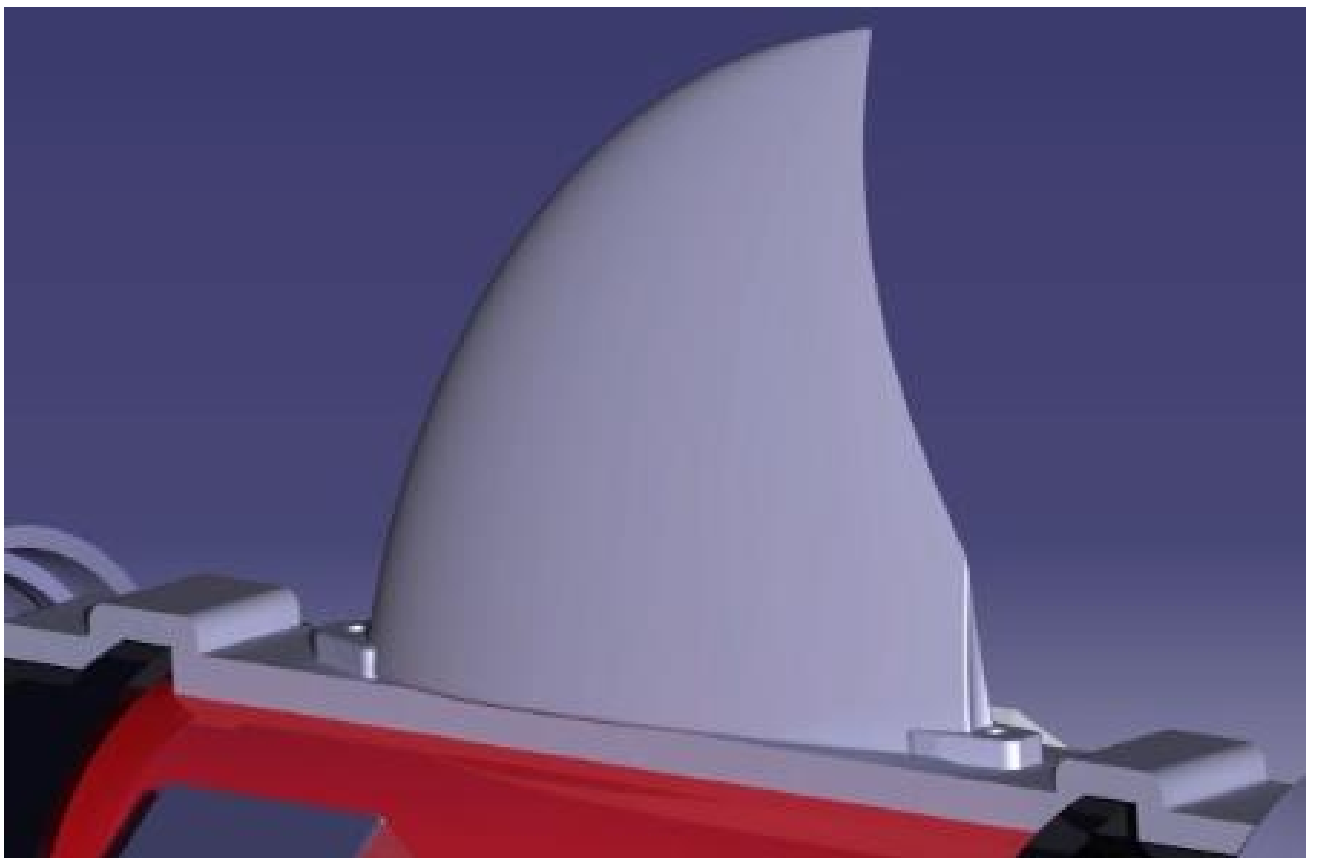
Source: Sfakiotakis *et al.* (1999)

Figura 3.3.2 - Diagramma del corpo libero e delle forze agenti

Delle tre pinne, create tutte sulla base del profilo NACA come spiegato nel capitolo 3.2, la pinna mediana è stata rimpicciolita di un terzo rispetto quella precedente così da rispettare maggiormente le proporzioni del nuovo prototipo. Non sono state apportate modifiche invece alla pinna di coda, riempita precedentemente col poliuretano espanso per essere chiusa a sezione garantendo di conseguenza una migliore galleggiabilità, essendo di un peso non trascurabile rispetto l'intero modello.

La pinna dorsale dell'immagine 3.3.3, è stata leggermente rimpicciolita; la sezione interna è stata resa ulteriormente cava rispetto a prima e sono stati modellati dei supporti alla base tali da essere compatibili con il porta carter che verrà montato sulla parte superiore del corpo del prototipo. Infine, la pinna dorsale di Borracciolo 2.0 della fig. 3.3.4, è stata modellata per poggiare direttamente sui supporti contro tappo-borraccia anziché sui carter universali, vista la diversa lunghezza di questi ultimi a causa della nuova forma del corpo e l'incompatibilità dei fori.

In particolare uno dei supporti contro tappo-borraccia, è stato modificato e reso più corto rispetto a quello classico usato in Borracciolo 1.0, come si può notare in figura 3.3.5, dato che non dovrà ospitare alcun contro tappo forato poiché la borraccia è ora semi-completa.



*Figura 3.3.3 - Particolare della pinna dorsale 1.0*



*Figura 3.3.4 - Particolare della pinna dorsale 2.0*



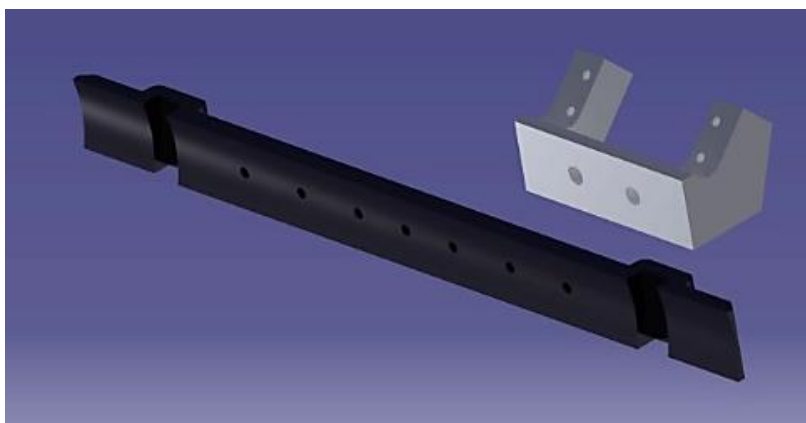
*Figura 3.3.5 - Supporto contro tappo-borraccia ridotto*

### 3.4 COMPARTI ACCESSORI

Il seguente paragrafo tratta tutte le parti accessorie di cui il robot è dotato, cioè:

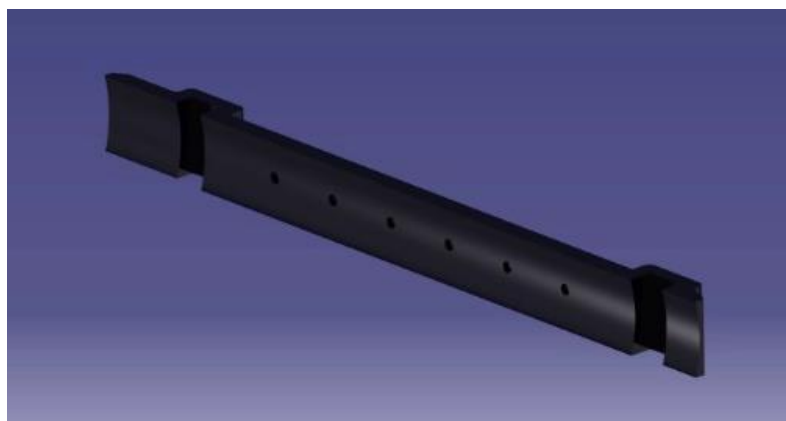
- Porta carter e carter per i servomotori;
- Collegamento motore pinna;
- Tappo forato della borraccia con relativo supporto;
- Contro coperchio con annesso aggancio gopro;
- Custodia impermeabile anteriore per gopro;
- Tappo Borracciolo 2.0.

I porta carter sono, come suggerisce il nome, dei supporti per ospitare i carter dei servomotori e sono stati resi universali aggiungendo una serie di fori pronti ad ospitare non solo ciò per cui sono stati creati, ma anche altri accessori che presentano la medesima foratura. Infatti sono compatibili anche con la pinna dorsale e sono stati stampati in VeroBlack. Siccome i servomotori SAVOX SW1211SG sono stati utilizzati in tutte le versioni del prototipo, si è ritenuto opportuno riusare i carter precedenti in VeroClear, senza apportare alcuna modifica.



*Figura 3.4.1 - Porta carter (in VeroBlack) e carter (in VeroClear) per il servomotore*

I porta carter laterali per Borracciolo 2.0, hanno una lunghezza minore in quanto sono stati modellati apposta per la nuova forma della borraccia, che in questo caso ha la parte posteriore completa. Come si può vedere in foto, hanno inoltre un foro in meno, cosa che permette comunque l'inserimento di altri componenti.



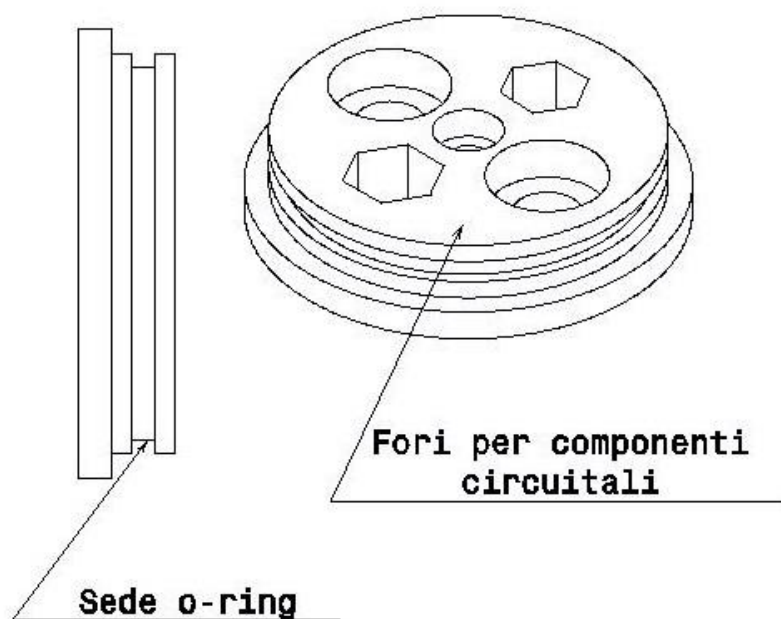
*Figura 3.4.2 - Porta carter più corto*

Altro componente riusato, è il cilindro di collegamento tra il disco in ABS di uscita del servomotore Savox e l'attacco delle pinne formato da un piccolo prisma a cilindro. Il pezzo è visibile nell'immagine sottostante.



*Figura 3.4.3 - Collegamento motore-pinna*

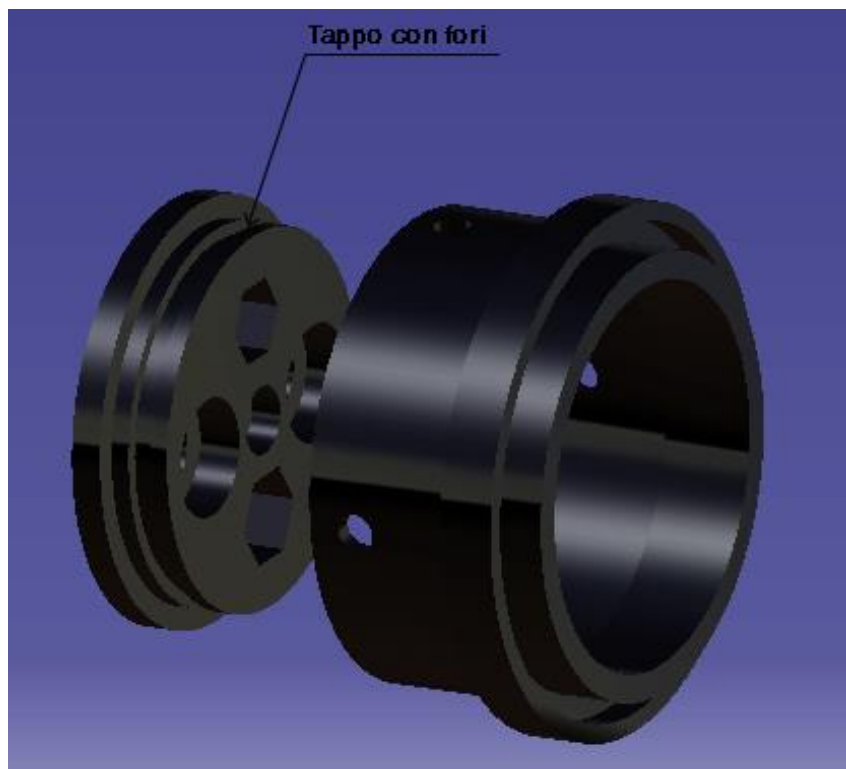
Anteriormente e posteriormente (nel caso di Borracciolo 2.0 solo anteriormente), è presente un tappo (figura 3.4.4) che presenta vari fori pronti ad ospitare la componente circuitale; la sua resistenza all'acqua è garantita da un **O-ring** opportunamente lubrificato ed inserito in un apposito alloggiamento, creando così, una guarnizione di tenuta. Inoltre, grazie alla sua naturale flessibilità, fintanto che la pressione dell'acqua proveniente dall'esterno non eccede lo stress da contatto sulle superfici della guarnizione, non possono esserci perdite.



*Figura 3.3.4 - Drafting tappo forato*

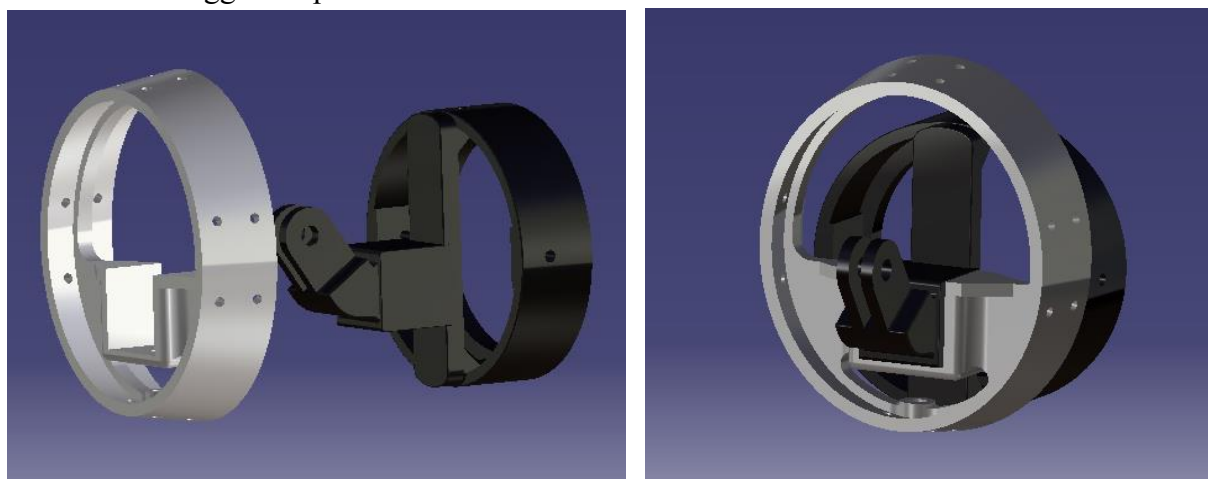


Anteriormente e posteriormente invece, è stato creato un supporto in VeroBlack che ospita la borraccia da un lato e il tappo che la chiude dall'altro.



*Figura 3.4.5 - Tappo con fori a sinistra e relativo supporto a destra*

Il contro tappo, stampato anch'esso in VeroBlack, è stato modellato in modo tale da essere un tutt'uno con il supporto che verrà montato nel telaio della vertebra, al quale è unito direttamente l'aggancio per la webcam.



*Figura 3.4.6 - Vertebra e contro tappo con supporto gopro prima e dopo l'assemblaggio*

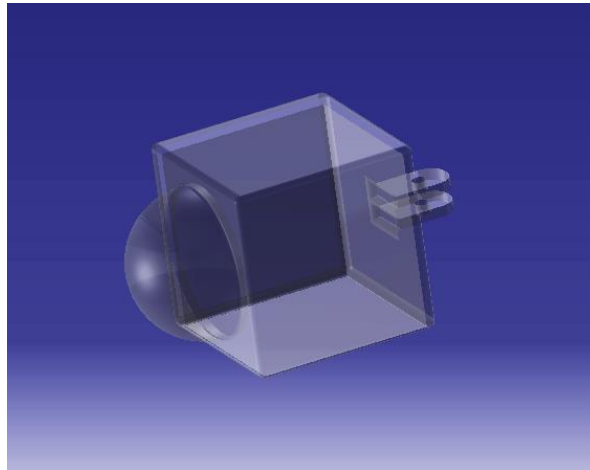
Il supporto gopro nella figura 3.4.6, è stato pensato affinché il suddetto dispositivo fosse centrato nell'asse centrale della vertebra e del corpo.



Inoltre, nella prua, è stata inserita una custodia impermeabile relativa all'action cam Kodak PixPro SP360 resistente ad immersioni fino a 60 metri di profondità (facilmente sostituibile con qualsiasi altra gopro con annessa custodia impermeabile). È stato necessario riprodurre il modello su CATIA per due ragioni:

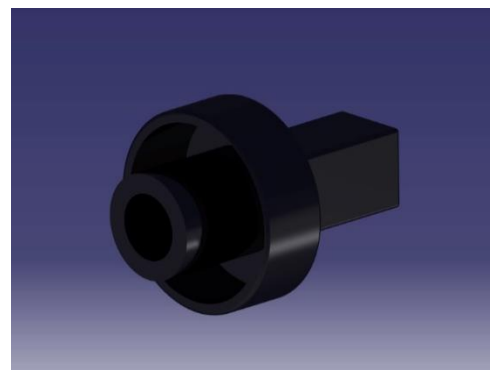
- La prima, è quella di avere un assieme completo e definitivo per tenere ben presente il pesce robotico completo, utile a chi in futuro si troverà a sviluppare versioni avanzate del prototipo;
- La seconda è che, essendo vuota, tiene al suo interno una certa quantità d'aria che sarà tenuta in considerazione da chi dopo di me e effettuerà studi relativi alla galleggiabilità, al centro di massa, alla spinta del pesce da parte dell'acqua ecc.

Tenendo conto di ciò, il modello della custodia è stato misurato e riprodotto nelle sue dimensioni su CATIA simulando un processo di reverse engineering sulla parte in esame.



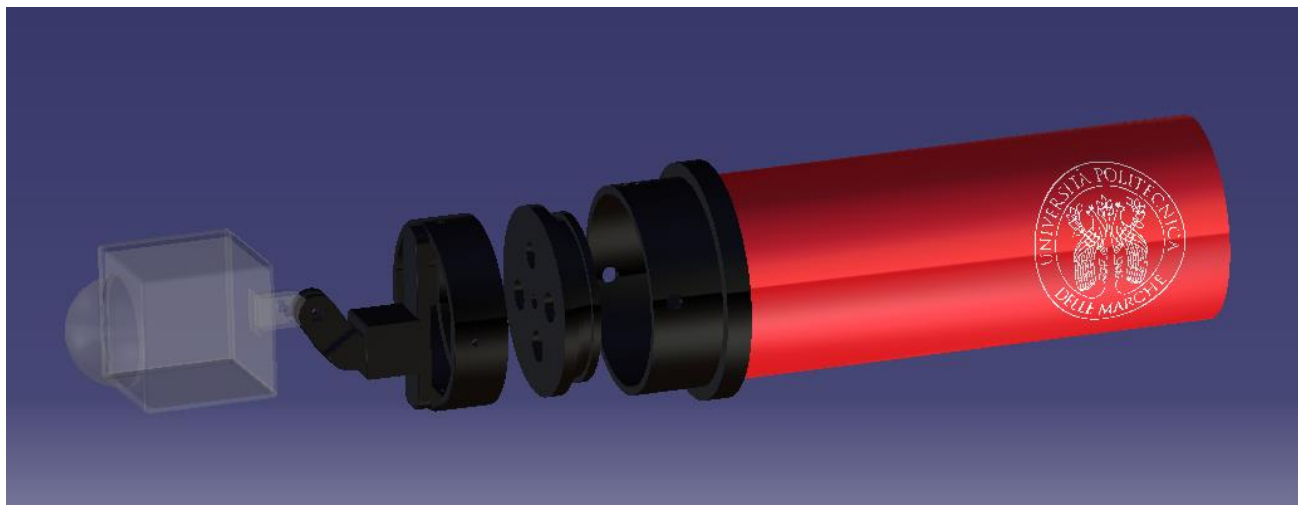
*Figura 3.4.7 - Custodia Kodak PixPro SP360*

Per finire, a poppa, in particolare per Borracciolo 2.0, è stato realizzato un altro componente, cioè un tappo che ha ad un'estremità un cilindro cavo che verrà filettato con una filettatrice gas  $\frac{3}{4}$ . Dopo essere stato avvitato, per garantire l'impermeabilità, verrà bloccato alla borraccia con la resina attraverso la giuntura. Nell'altra estremità presenta un cubo che andrà ad unirsi alla vertebra posteriore.

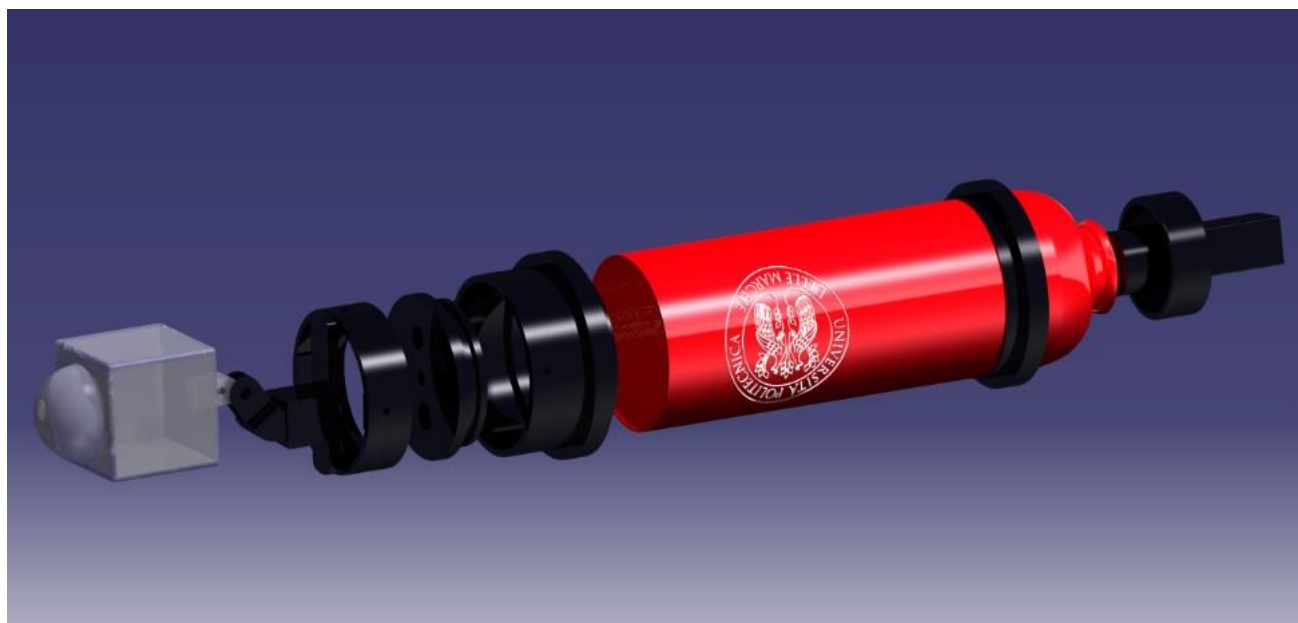


*Figura 3.4.8 - Tappo con giuntura Borracciolo 2.0*

Di seguito due rappresentazioni di come sono stati assemblati tutti i componenti finora citati insieme al corpo principale, rispettivamente per Borracciolo 1.0 e Borracciolo 2.0.



*Figura 3.4.9 - Sequenza del montaggio dei vari componenti con il corpo di Borracciolo 1.0*

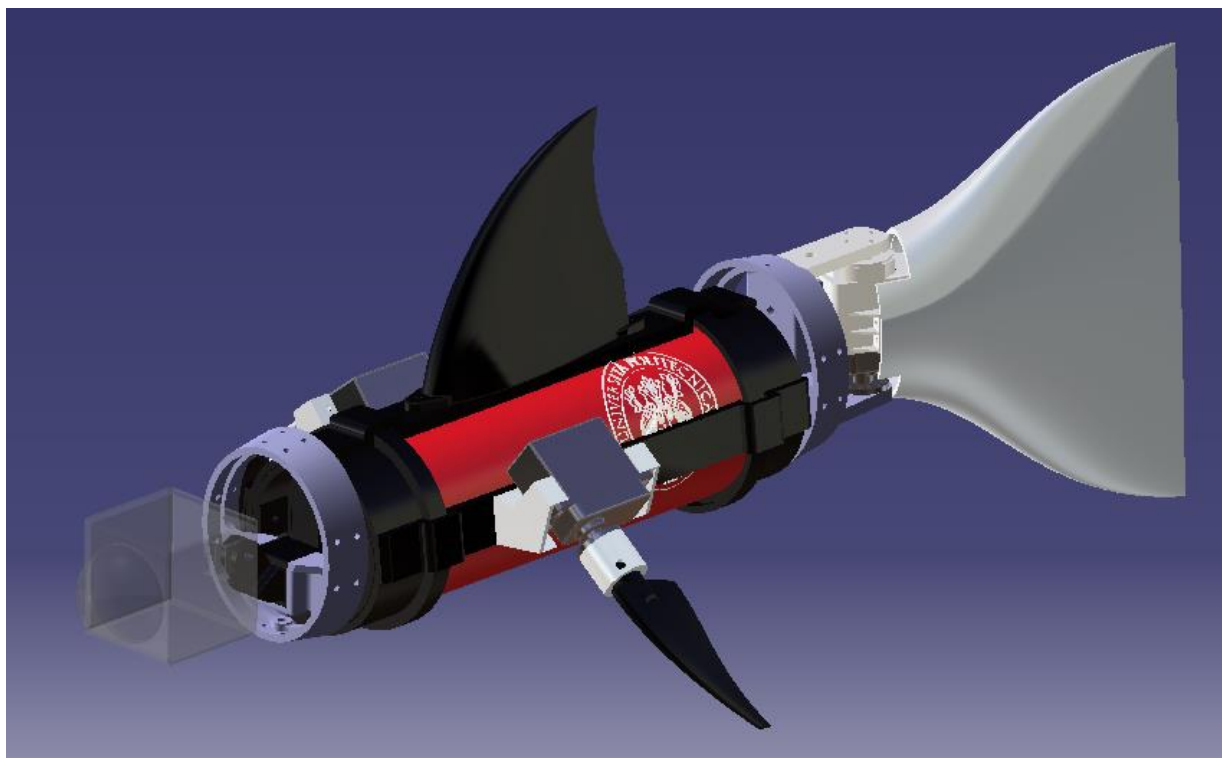


*Figura 3.4.10 - Sequenza del montaggio dei vari componenti con il corpo di Borracciolo 2.0*

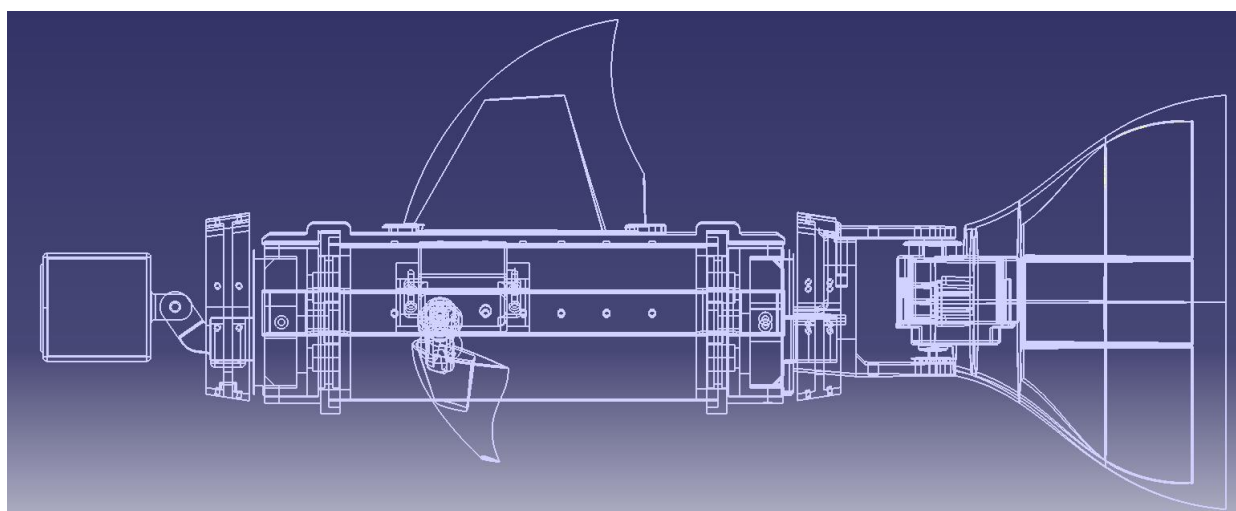
Per testare infine l'impermeabilità dopo l'assemblaggio di tutto, è stata usata una pompa a vuoto, che simula la pressione presente nel mare a più atmosfere e quindi a svariate profondità.

### 3.5 ASSIEME E DISTINTA BASE

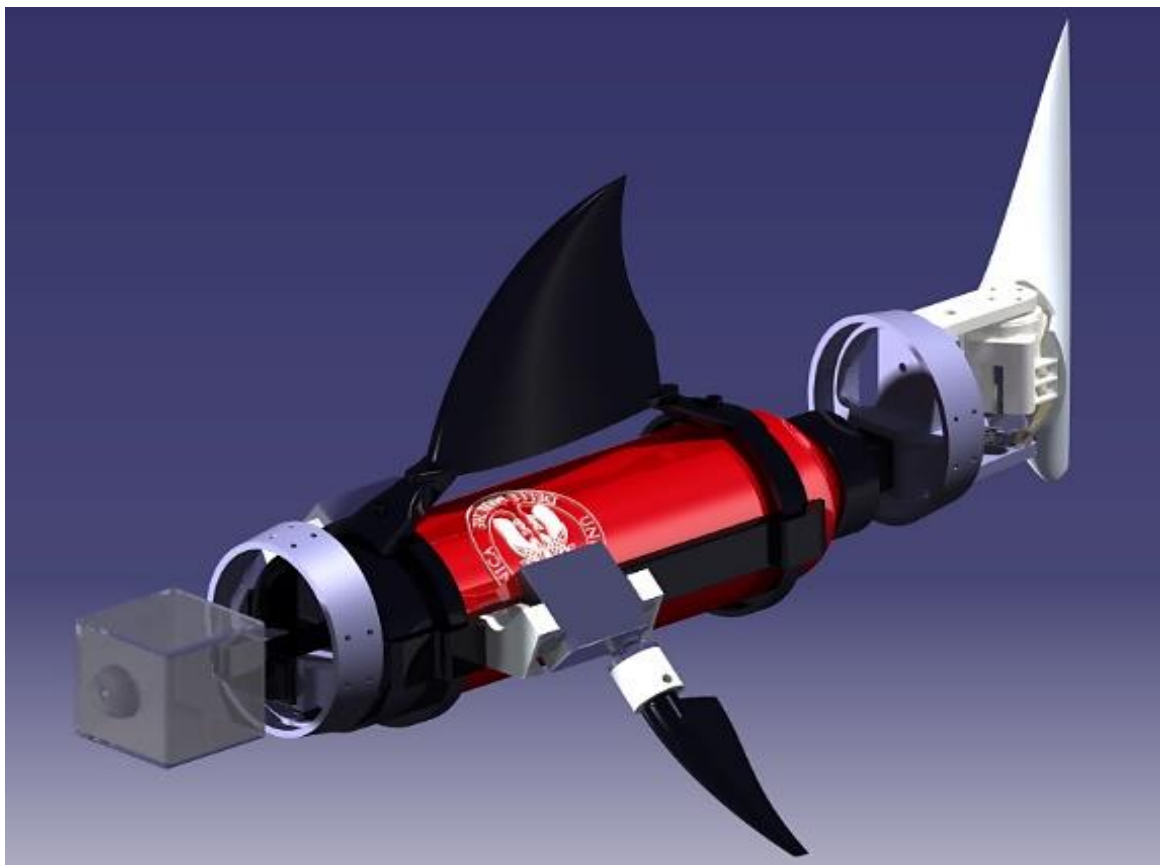
Alla luce del lavoro di modellazione svolto e mostrato nei paragrafi precedenti di questo capitolo, si riportano alcune immagini dell'assieme realizzato e del modello effettivamente creato rispettivamente di Borracciolo 1.0 e Borracciolo 2.0, anche in NHR per avere una visione complessivamente migliore. Seguirà poi una lista completa dei materiali utilizzati per la realizzazione del prototipo.



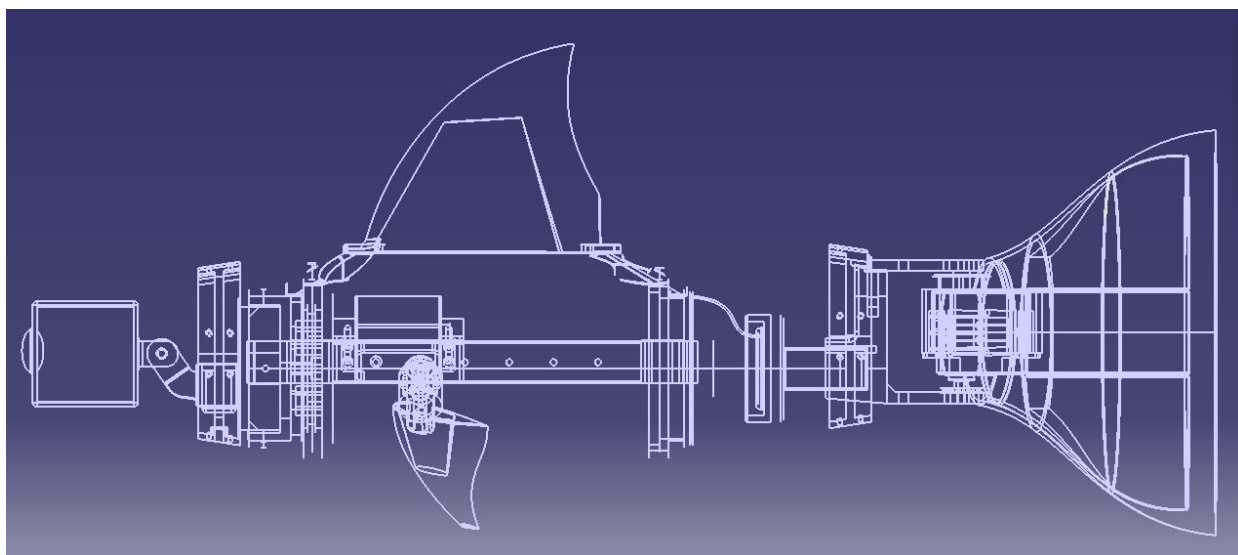
*Figura 3.5.1 - Assieme di Borracciolo 1.0*



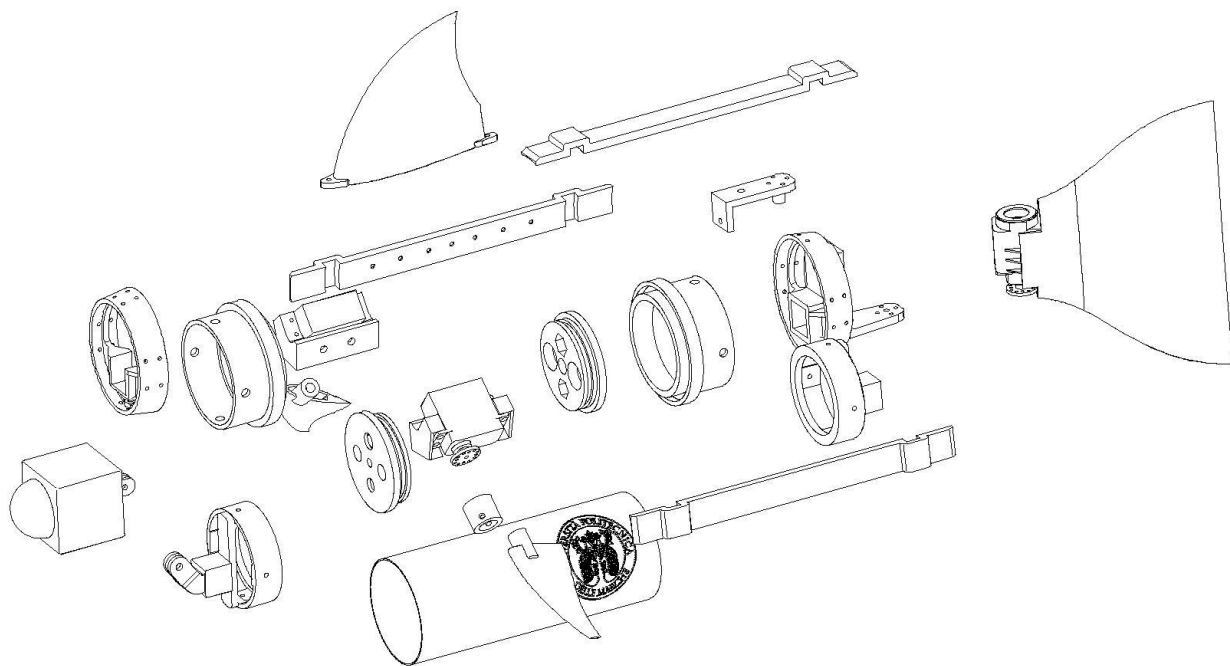
*Figura 3.5.2 - Vista laterale in NHR (Wireframe)*



*Figura 3.5.3 – Assieme di Borracciolo 2.0*



*Figura 3.5.4 - Vista laterale in NHR (Wireframe) Borracciolo 2.0*

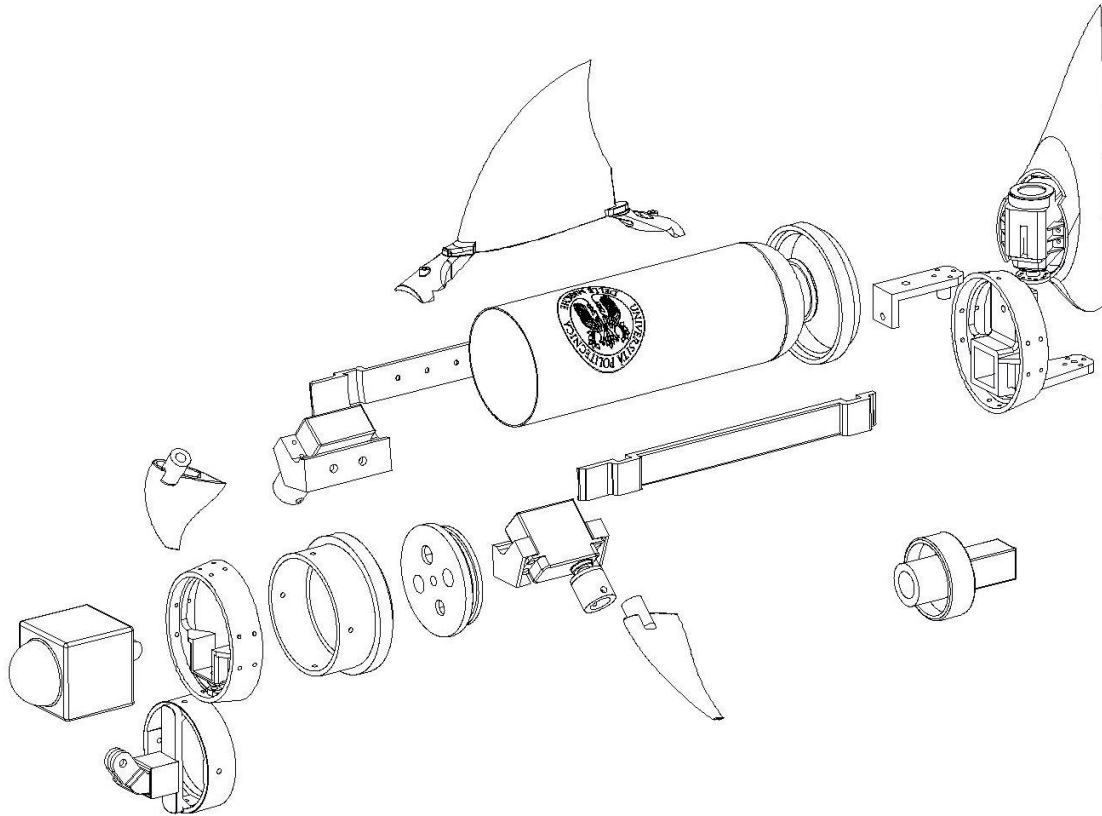


*Figura 3.5.5 - Esploso del prototipo Borracciolo 1.0*

La seguente distinta base (Bill Of Materials – BOM) mostra l’elenco dei componenti necessari per la realizzazione di questo prototipo.

<u><b>Oggetto</b></u>	<u><b>Caratteristiche</b></u>	<u><b>Quantità</b></u>
Cover Camera Kodak	Plexiglass	1
Contro tappo con supporto gopro	VeroClear	1
Vertebra anteriore	VeroBlack	1
Supporto contro tappo-borraccia	VeroBlack	2
Tappo con fori	VeroBlack	2
Borraccia UNIVPM	Metallo	1
Porta carter universale	VeroBlack	3
Carter savox pinna mediana	VeroClear	2
Disco per savox	ABS	2
Cilindro motore-pinna	VeroClear	2
Savox	Alluminio	3
Pinna mediana	VeroBlack	2
Pinna dorsale 1.0	VeroBlack	1
Contro tappo posteriore senza supporto gopro	VeroBlack	1
Vertebra posteriore	VeroClear	1
Basetta 5.0	VeroClear	1
Pinna caudale	VeroClear	1

*Figura 3.5.6 - Distinta base 1.0*



*Figura 3.5.7 - Esploso del prototipo Borracciolo 2.0*

La seguente distinta base (Bill Of Materials – BOM) mostra l’elenco dei componenti necessari per la realizzazione di questo prototipo.

<u><b>Oggetto</b></u>	<u><b>Caratteristiche</b></u>	<u><b>Quantità</b></u>
Cover Camera Kodak	Plexiglass	1
Contro tappo con supporto gopro	VeroClear	1
Vertebra anteriore	VeroBlack	1
Supporto contro tappo-borraccia	VeroBlack	1
Tappo con fori	VeroBlack	1
Borraccia UNIVPM semicompleta	Metallo	1
Porta carter universale più corto	VeroBlack	2
Carter savox pinna mediana	VeroClear	2
Disco per savox	ABS	2
Cilindro motore-pinna	VeroClear	2
Savox	Alluminio	3
Pinna mediana	VeroBlack	2
Pinna dorsale 2.0	VeroBlack	1
Supporto contro tappo-borraccia ridotto	VeroBlack	1
Tappo con giuntura	VeroBlack	1
Vertebra posteriore	VeroClear	1
Basetta 5.0	VeroBlack	1
Pinna caudale	VeroClear	1

*Figura 3.5.8 - Distinta base 2.0*

## 4. APPLICAZIONE ANALISI FEM

### 4.1 ACCORTEZZE NELLA MODELLAZIONE DEI COMPONENTI PER DIMINUIRE LA DEFORMAZIONE

Come introdotto nel paragrafo 3.3, è stato usato il software CATIA anche per porre particolare attenzione alla modellazione di determinati componenti della sezione posteriore del prototipo che sono sottoposti continuamente a determinati sforzi.

Non essendo il principale obiettivo della tesi, quello che si è fatto è stato diminuire l'effetto della forza agente in un determinato punto di un componente, così che a lungo andare, si possa ridurre al minimo il danno causato dalla deformazione meccanica. È stata svolta quindi un'analisi qualitativa impostando una simulazione il più vicino possibile alla realtà. La parte analizzata è stata il contro coperchio posteriore che ha sede nella vertebra direttamente collegata al servo motore che muove la pinna caudale ed è quindi sottoposto a molti sollecitamenti; di conseguenza mi sono limitato a correggerlo aggiungendo una nervatura per limitare il carico presente e cercare di ridurre al minimo la possibilità di un'eventuale rottura.

Ai fini dell'analisi, le deformazioni verranno considerate in campo prettamente elastico; quindi viene ipotizzato che il corpo torni nella posizione originale al cessare dei carichi agenti, con un modulo di Young del tipo:

$$E = \frac{\sigma}{\varepsilon}$$

dove la  $\sigma$  non supera il carico specifico di snervamento, oltre il quale il corpo smette di in modo elastico. La regione del campo elastico è descritta nella figura sottostante ed ha come valore limite il punto  $(S_0; e_0)$ .

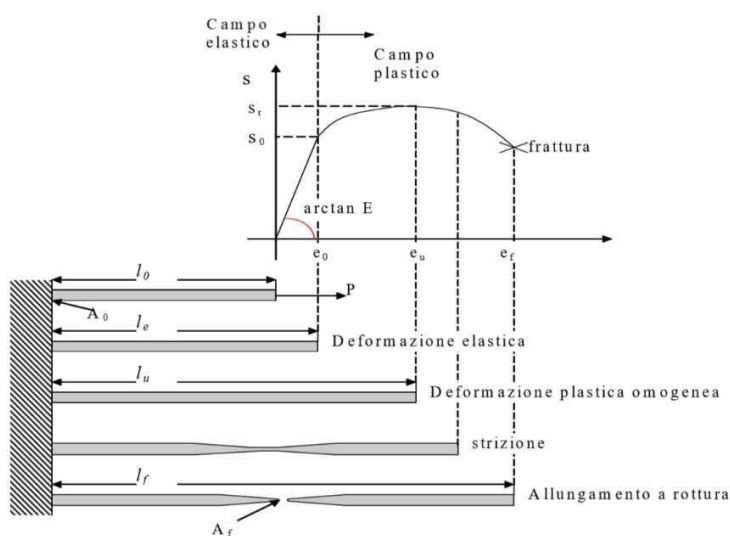


Figura 4.1 - Curva Tensione-Deformazione

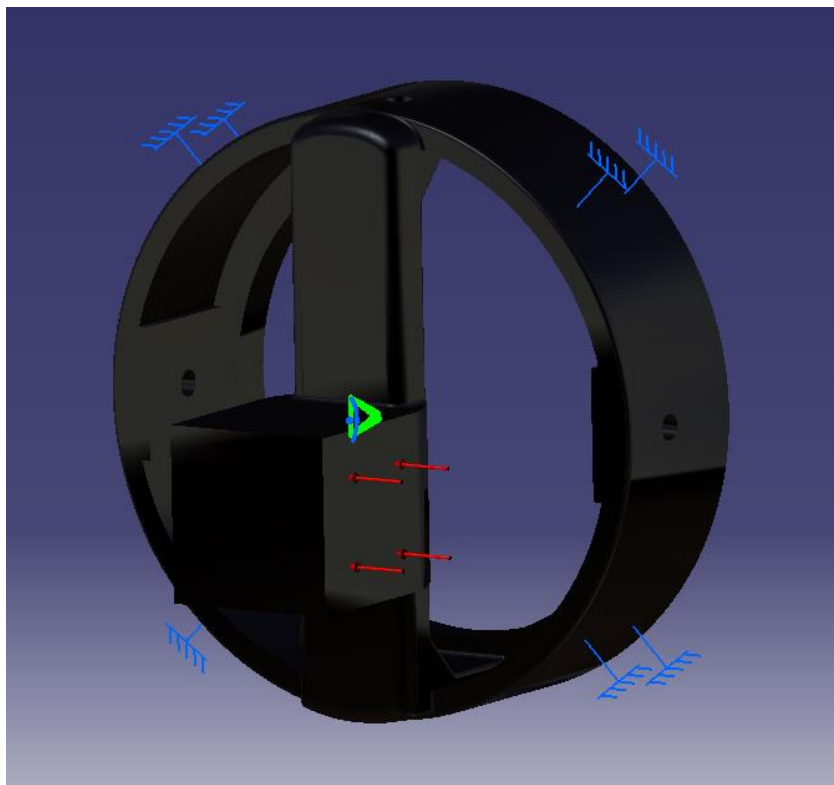


## 4.2 PROCEDIMENTO EFFETTUATO

Per avviare l'analisi statica e il calcolo della soluzione alla deformazione totale, si inizia inserendo le caratteristiche del materiale usato, il VeroBlack Plus, presenti nel datasheet di Stratasys che risultano essere:

- $\rho = 1175 \frac{Kg}{m^3}$
- $E = 2500 GPa$
- $\nu = 0,35$

Successivamente bisogna creare una mesh (griglia) del modello composta da primitive. Si è scelto di optare per la forma tetraedrica parabolica, con una risoluzione di 4.3 mm ad elemento, poiché parte del pezzo trattato è circolare e si ottengono così risultati più precisi. Dopo aver vincolato il corpo nei suoi gradi di libertà impostando l'incastro nella parte esterna (simulando la sua posizione nel supporto contro tappo-borraccia) e aver inserito una pressione esercitata su un lato del cubo che viene inserito nella vertebra, di un valore indicativo di  $40 \frac{N}{m^2}$ , si può procedere con il calcolo di tutti i parametri da parte del software.



*Figura 4.2.1 - Applicazione dei carichi (in rosso) e dei vincoli (in blu)*

La visualizzazione delle tensioni elastiche è data dall'Equivalent Elastic Strain (Von Mises), presente nelle seguenti immagini.



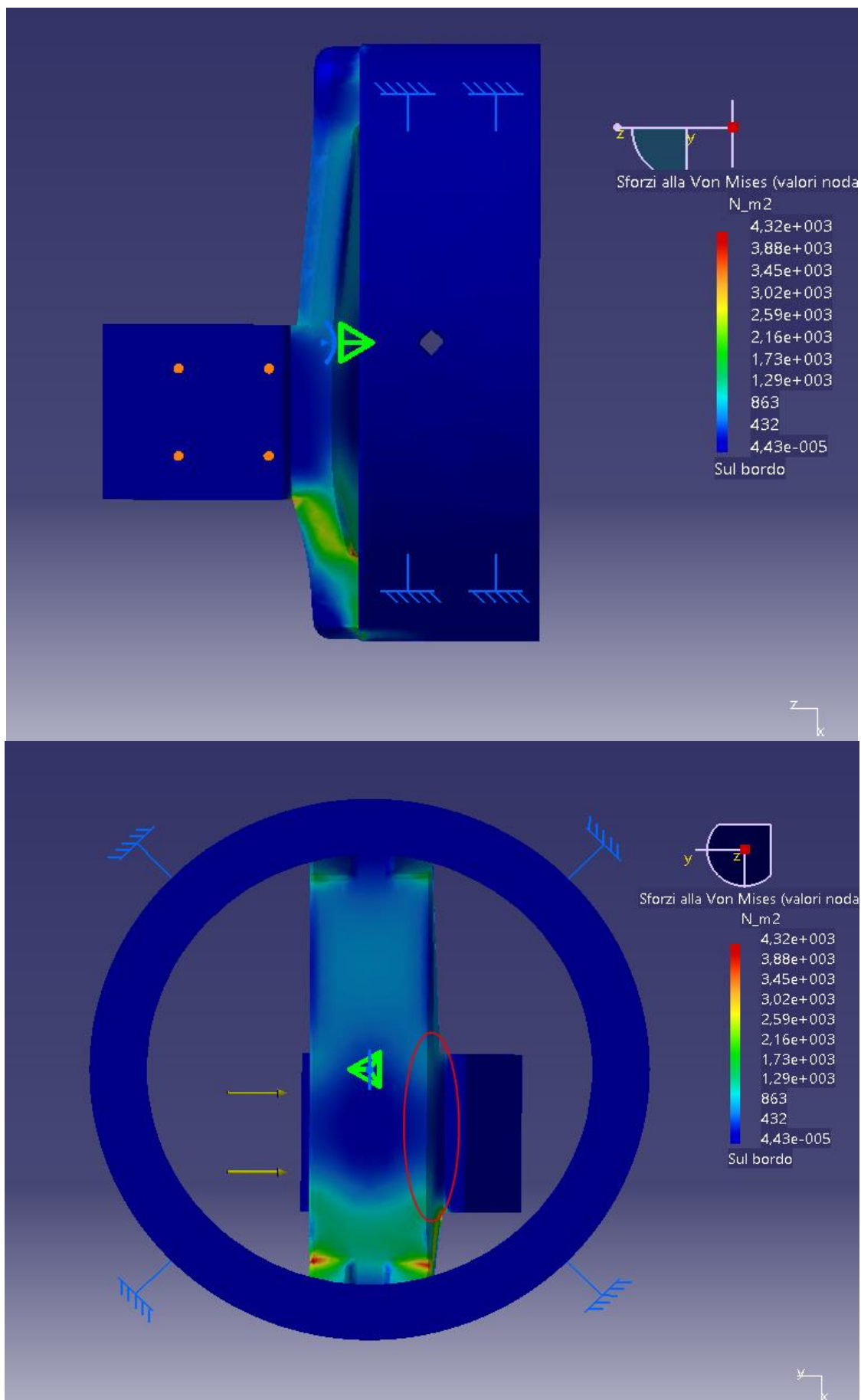


Figura 4.2.2 Prima simulazione della deformazione totale

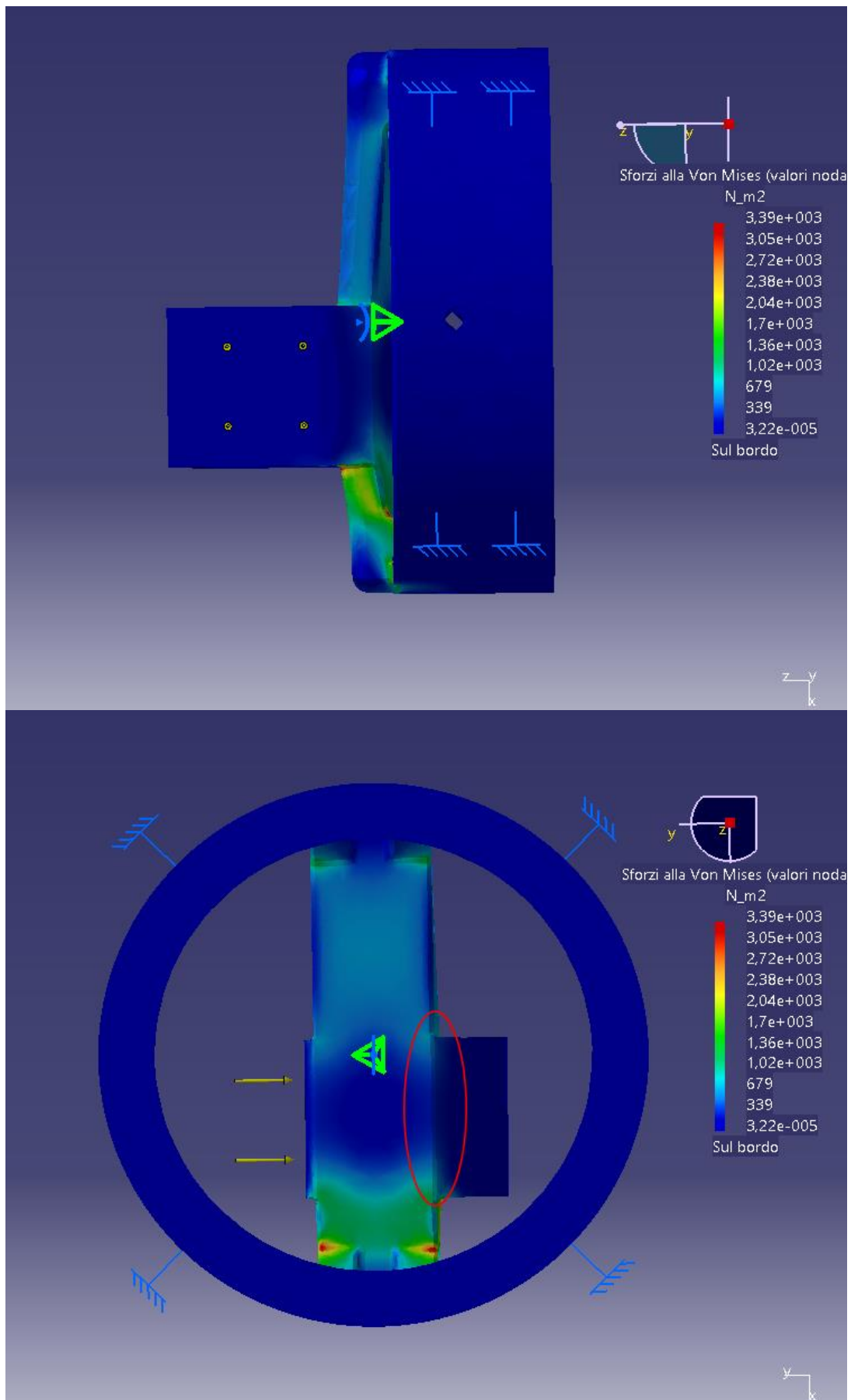


Figura 4.2.3 Seconda simulazione della deformazione totale

Nella prima simulazione, è presente il contro-tappo senza la nervatura. Si può constatare che il valore massimo dello sforzo al quale è sottoposto il pezzo è di circa  $4320 \frac{N}{m^2}$ .

Nella seconda simulazione invece, il contro-tappo è provvisto di nervatura lungo il lato del cubo da entrambe le parti. Il valore massimo dello sforzo in questo caso è  $3390 \frac{N}{m^2}$ .

I numeri esposti mostrano la diminuzione della sollecitazione massima che avviene sul pezzo. Anche nell'area evidenziata dall'ellisse, è minore la presenza dell'effetto della forza e in generale si può notare una flessione minore verso destra della trave con il cubo annesso. Di conseguenza si è ritenuto opportuno scegliere il secondo pezzo, ottenendo così una maggiore resistenza del prototipo.

L'esame qui illustrato non è stato svolto per il contro-tappo anteriore poiché non è sottoposto a nessun tipo di forza tale da provocare deformazioni o rotture.

## 5. SISTEMA DI NAVIGAZIONE E GUIDA

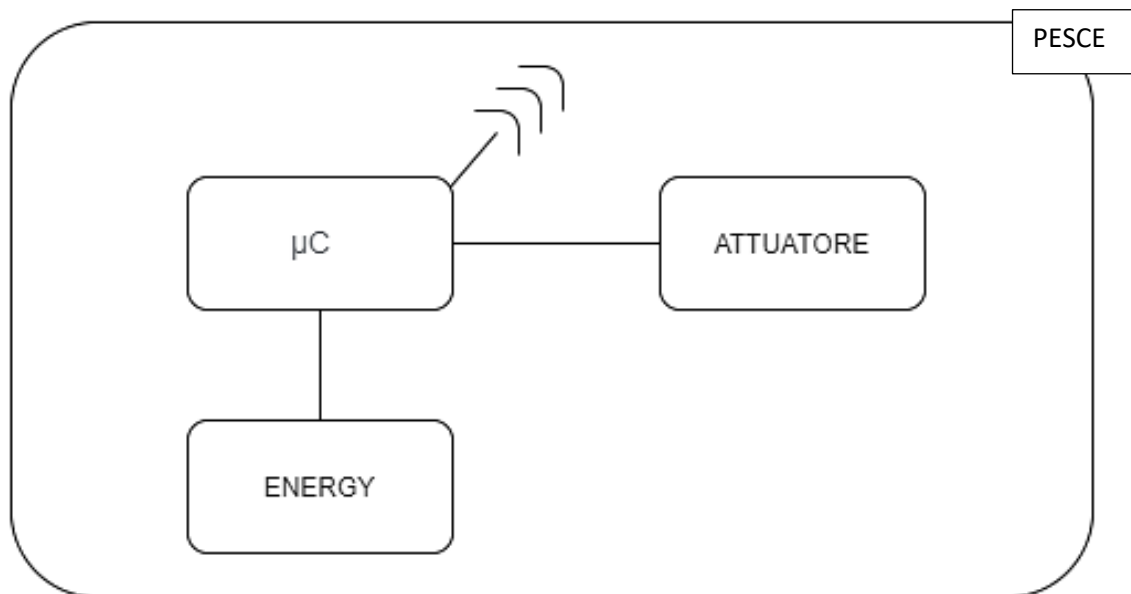
Completato il disegno mecatronico in cui si ha il connubio di tipo meccanico/attuatore, è stato studiato il sistema di navigazione e guida. Quest'ultimo può essere rappresentato attraverso l'uso di due schemi ben precisi. Uno, in figura 5.1, è rappresentato dal pesce vero e proprio, che è composto da:

- un microcontrollore;
- un attuatore;
- l'alimentazione.

L'MCU (MicroController Unit) è un dispositivo elettronico integrato che utilizza tecniche di microelettronica per ridurre in un piccolo package vari componenti quali CPU (Central Processing Unit) e memoria. Presenta diversi pin di ingresso e uscita, attraverso i quali è possibile interagire con il mondo esterno. Il microcontrollore necessita di una programmazione attraverso un insieme di istruzioni che rappresenta il software (o l'app in questo caso) installato sulla memoria del dispositivo. Il principale impiego è rivolto ai sistemi embedded come questo, ovvero per tutte quelle applicazioni specifiche relative al controllo digitale. Nel suddetto contesto, sarà indispensabile ai fini del controllo del pesce robot.

L'attuatore è un dispositivo che trasforma una decisione automatica di comando, elaborata dall'MCU, in un'azione fisica sul processo oggetto della regolazione. L'insieme degli attuatori rappresenta quindi la 'muscolatura' del sistema di automazione che compone tutto il progetto, laddove il microcontrollore rappresenta il cervello e il software di controllo l'intelligenza. Gli attuatori meccanici, per la maggior parte costituiti dai motori elettrici e a combustione interna, sono in grado di convertire i comandi di controllo in coppie o forze meccaniche che sono richieste dall'unità elettronica di controllo.

L'alimentazione di 7.2V è il mezzo attraverso il quale si darà vita al prototipo.



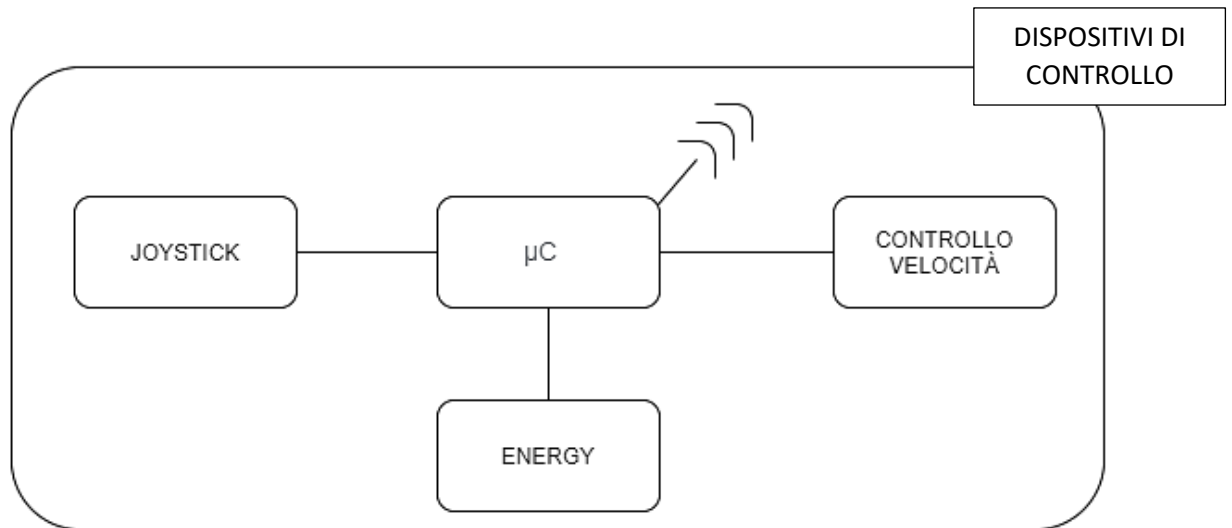
*Figura 4.1 - Schema del sistema di navigazione*

Il secondo schema invece, rappresentato in figura 5.2, è composto da:

- un microcontrollore;
- un potenziometro;
- un joystick;
- l'alimentazione.

Il potenziometro è un dispositivo che agisce come partitore di tensione variabile. Converte il movimento rotatorio di chi lo utilizza, in un cambiamento di resistenza. La struttura interna del potenziometro è uguale, in linea di massima, per tutti i tipi in commercio. Consiste in una resistenza fissa collegata a due terminali di contatto su cui si sposta un contatto mobile, azionato da un comando esterno, che può percorrerla da un estremo all'altro. Questo contatto è collegato a un terzo terminale. In questo modo, si può ottenere una resistenza del valore desiderato tra uno degli estremi del potenziometro e il contatto mobile. Dal punto di vista digitale, ciò che accade attraverso l'uso del microcontrollore è la conversione del segnale da analogico a digitale, che va da 0 a 1024. In base a quest'ultimo, varia appunto la quantità di tensione che viene applicata.

Il joystick è una periferica che trasforma i movimenti di una leva manovrata dall'utente in una serie di segnali elettrici o elettronici che permettono di controllare in questo caso i tre attuatori meccanici utilizzati nel prototipo.

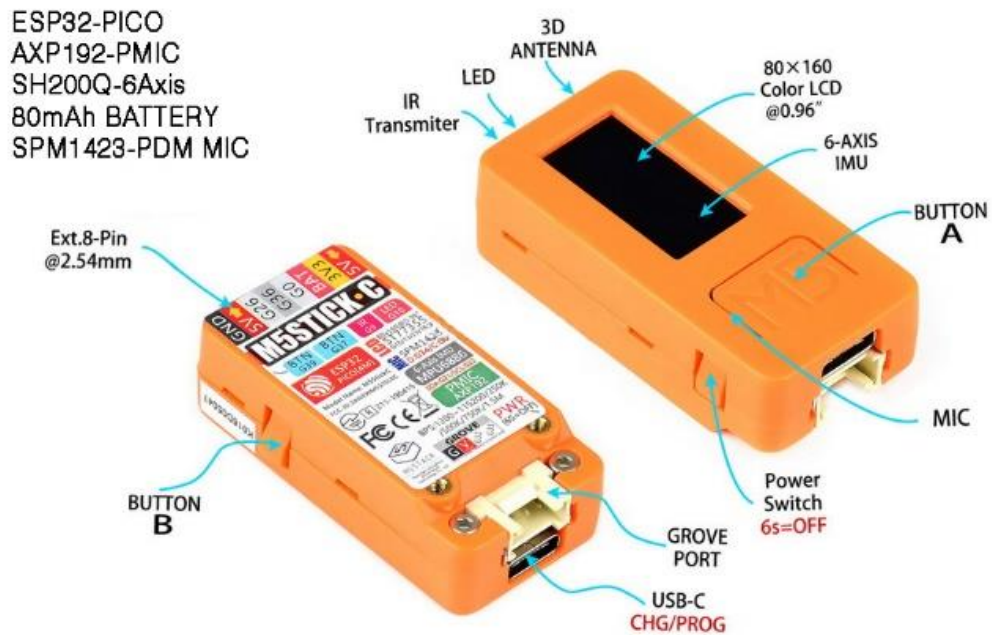


*Figura 5.2 - Schema del sistema di guida*

Per mettere in comunicazione i due sistemi senza utilizzare il wifi, si è optato per il protocollo sviluppato da Espressif, ESP-NOW. Per sviluppare tutto ciò di cui è stato discusso fin'ora, si è scelta la tecnologia M5StickC perché facile da programmare ai fini della robotica educativa finalizzata ai bambini, che è ciò per cui è nato il presente elaborato.

## 5.1 M5STICKC

M5StickC ESP32 Development Board è un microcontrollore dalle dimensioni ridotte estremamente versatile, programmabile in diverse modalità e permette di realizzare facilmente e rapidamente progetti IoT. Utilizza il chip ESP32 che offre connettività WiFi e Bluetooth, un accelerometro a 6 assi, un led integrato, una batteria da 80mAh e una memoria flash da 4Mb.



*Figura 5.1.1 – Vista A M5StickC*



*Figura 5.1.2 Vista B M5StickC*

Sul corpo del dispositivo, si trovano i due pulsanti A e B che hanno varie funzioni, un interruttore e un display LCD.

Questo dispositivo in miniatura è utile per creare un semplice dispositivo di controllo remoto o un progetto più sofisticato. I bambini, lo troveranno un ottimo strumento per imparare a programmare. Inoltre grazie all'interfaccia GROVE e GPIO è possibile connettere dei dispositivi modulari esterni di vario genere. Sono stati utilizzati in questo caso il PuppyC, il potenziometro e il joystick.



Figura 5.1.3 PuppyC

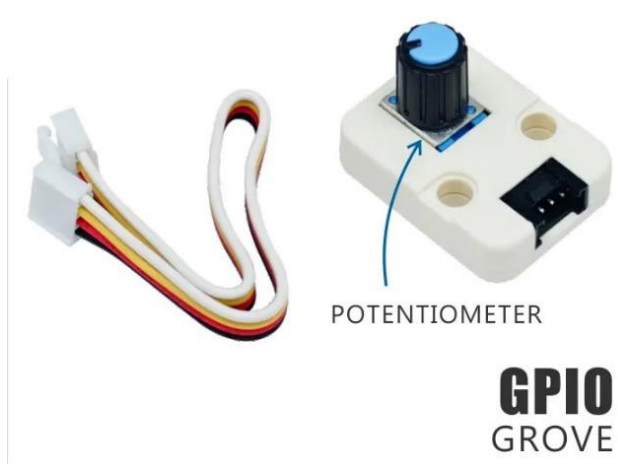


Figura 5.1. 4 Potenziometro



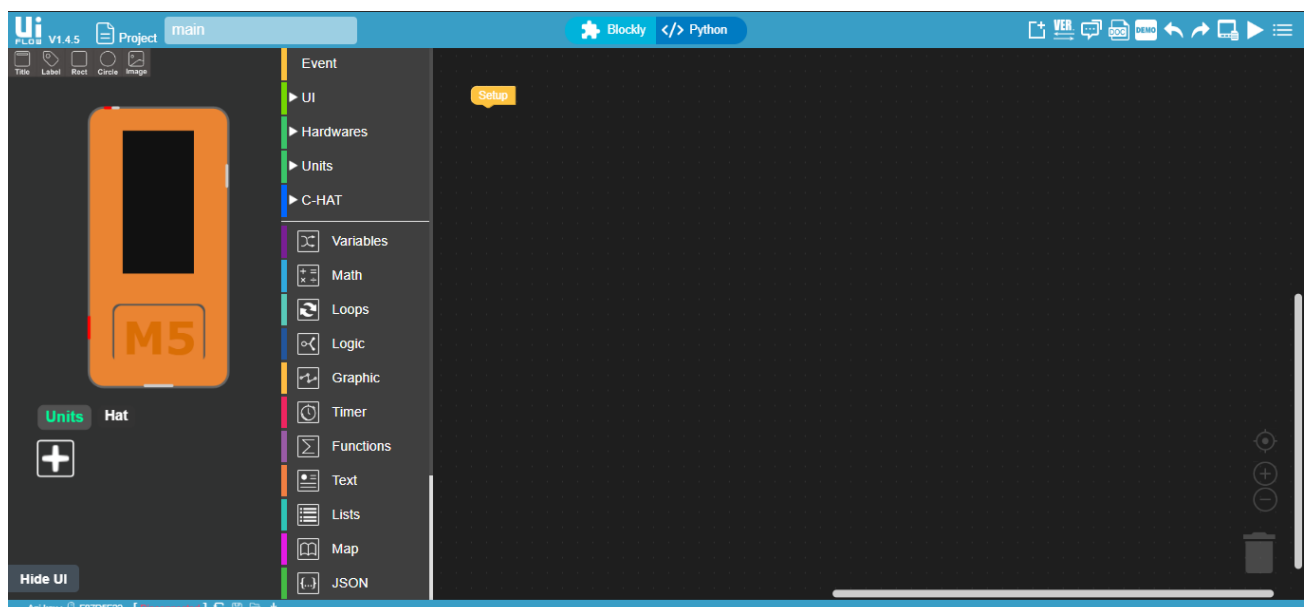
Figura 5.1.5 Joystick

Per poter programmare tutti i dispositivi sopra elencati, è stato utilizzato l'IDE UiFlow descritto nel prossimo paragrafo.



## 5.2 UIFLOW

UIFlow è una piattaforma di programmazione appositamente progettata per i dispositivi M5Stack. L'IDE di programmazione grafico a blocchi è basato su Blockly, linguaggio grafico di programmazione ben conosciuto in campo didattico. UIFlow consente inoltre di programmare qualsiasi oggetto M5Stack in MicroPython, implementazione di Python 3 per microcontrollori e sistemi embedded. MicroPython, è un linguaggio di programmazione snello ed efficiente ideale per attività di coding a scuola e da sempre impiegato in campo scientifico. UIFlow fornisce le funzionalità necessarie per la realizzazione, in maniera estremamente semplice, di progetti con forte interazione con il mondo reale, ideale quindi per chi si avvicina al mondo della prototipazione elettronica, dell'automazione e della programmazione.



*Figura 5.2.1 - Interfaccia IDE UIFlow*

L'IDE UIFlow può essere utilizzato on-line su browser oppure localmente scaricandolo dal sito principale. A seconda di come lo si vuole utilizzare, la programmazione del microcontrollore attraverso il presente IDE, può essere effettuata in due modi, via usb (locale) e via wifi (browser). Per quanto riguarda la connessione via usb, bisogna innanzitutto tenere premuto il pulsante di accensione dell'M5StickC per 2 secondi. Non appena compare il logo “UiFlow”, bisogna premere sul tasto A per accedere al menu impostazioni e con l'ausilio del pulsante B, ci si può muovere tra le varie opzioni fino ad arrivare alla voce “Setup” e scegliere “USB mode”.

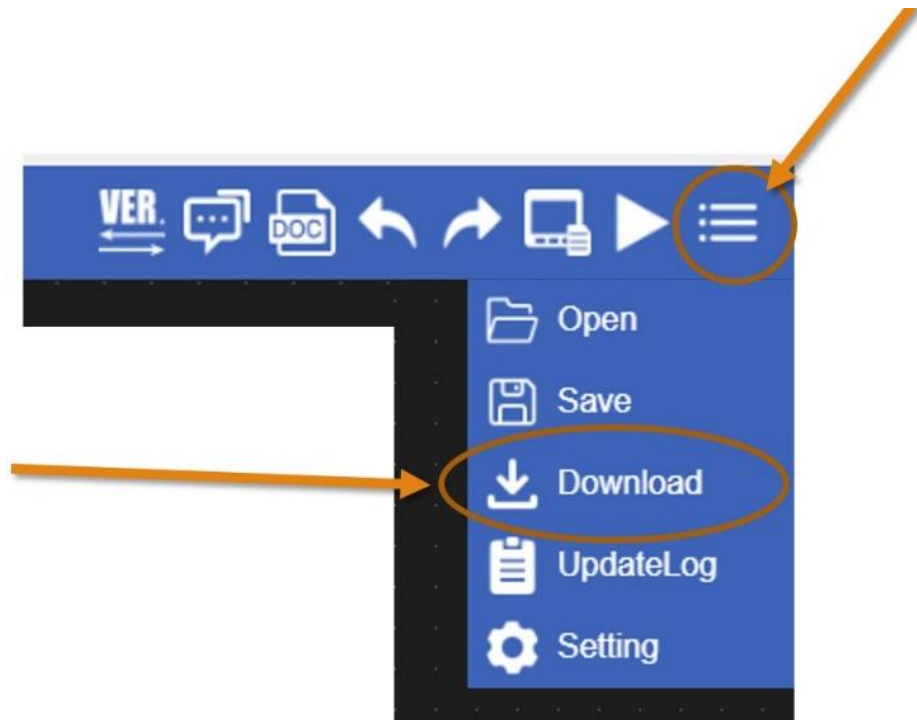


*Figura 5.2.2 - Settaggio USB*

Fatto ciò, bisogna stabilire una connessione tra i due terminali collegando semplicemente attraverso l'usb type-c l'M5StickC al pc. Successivamente, è possibile scegliere, dopo aver cliccato in basso a sinistra della finestra dell'IDE locale, la porta usb seriale in cui è stata inserita la periferica. Nel caso in cui l'M5 non dovesse essere riconosciuto dal sistema, basta scaricare i driver ufficiali presenti nel sito dell'M5Stack. Dopo aver creato il programma, per scaricarlo sul dispositivo, basta cliccare sul tasto Run situato in alto a destra. Apparirà un messaggio di conferma per l'avvenuto upload del codice.

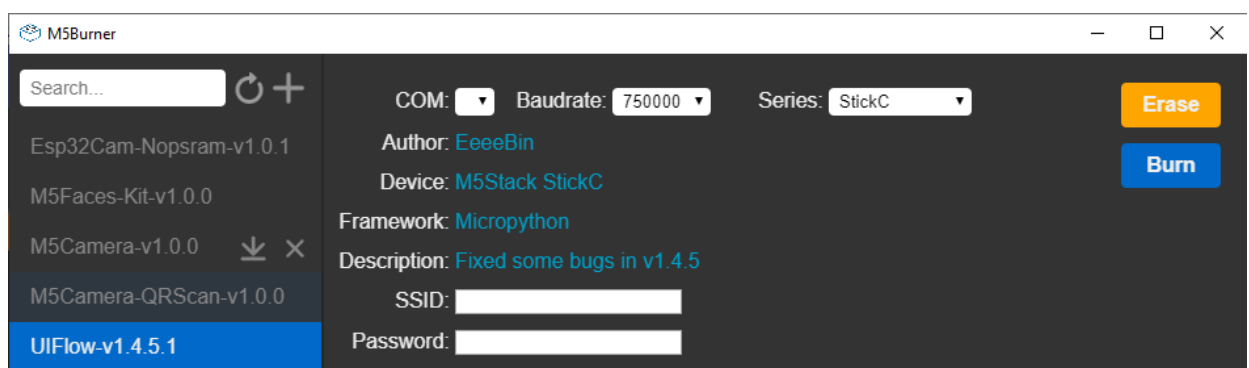
Per quanto riguarda la programmazione wifi invece, sull'M5StickC bisogna fare la medesima operazione, solo che si deve scegliere "Cloud mode" anziché "USB mode". Quindi è necessario collegare il microcontrollore alla stessa rete alla quale è connesso il dispositivo dal quale viene effettuata la programmazione (pc, tablet, smartphone ecc). Successivamente, verrà visualizzato l'API da inserire nell'IDE browser questa volta, così da poter procedere con la programmazione.

Dopo aver completato la programmazione, per inviare il codice all'M5, bisogna andare sul menu in alto a destra e cliccare sulla voce "Download". Il dispositivo verrà resettato e verrà avviato da quel momento in poi sempre con l'app caricata.



*Figura 5.2.3 - Upload attraverso il browser*

Nel caso in cui l'M5StickC non possedesse il firmware, bisogna installarlo. Per farlo è necessario aprire l'IDE UIFlow locale e scegliere sulla barra dei menu "FirmwareBurner". Dopodiché bisogna scegliere la versione da installare (per esempio UIFlow 1.4.5.1 che è la versione usata durante questo progetto), la porta usb alla quale è collegato il dispositivo, il baudrate (solitamente la migliore frequenza è 115200 o 921600) ed infine il prodotto specifico tra quelli che presenta M5Stack. Per un'installazione pulita, bisogna prima eliminare un eventuale versione precedente di UIFlow e poi avviare la scrittura del nuovo software. Se si intende collegarsi ad una rete wifi, si possono inserire le credenziali di accesso al momento dell'installazione del software attraverso i campi "SSID" e "Password". Di seguito una schermata di ciò che compare nel programma di installazione.



*Figura 5.2.4 - M5Burner*

### 5.3 PRIMA ESPERIENZA CON PUPPYC

Il PuppyC permette tutti i movimenti delle pinne del pesce. Un primo test è stato effettuato controllando il puppy con il pc, facendo eseguire dei comandi semplici a intervalli di tempo regolari.

Successivamente sono stati collegati quindi all'M5StickC il puppy attraverso la porta GPIO (hat) e il potenziometro nella porta GROVE (unit).

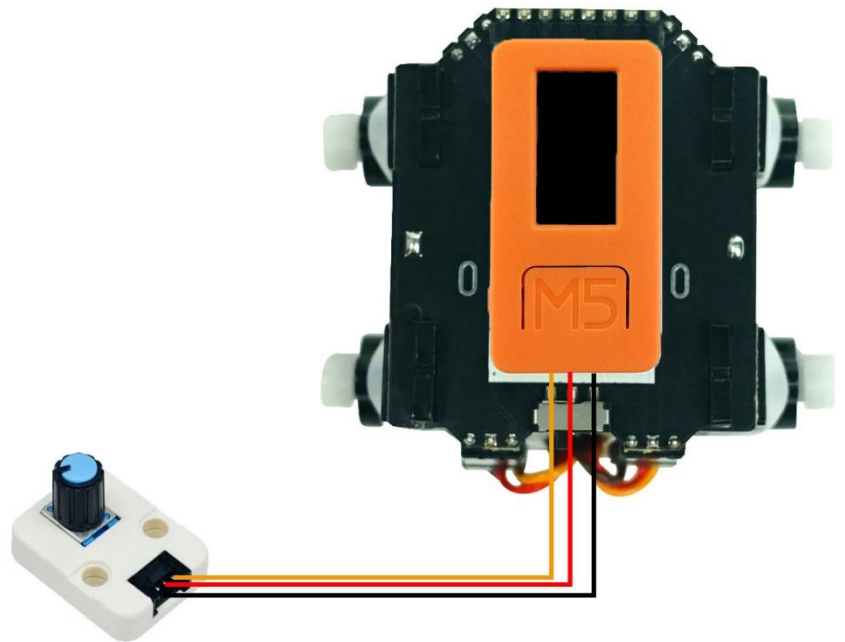


Figura 5.3.1 - Schema circuitale PuppyC-potenzimetro

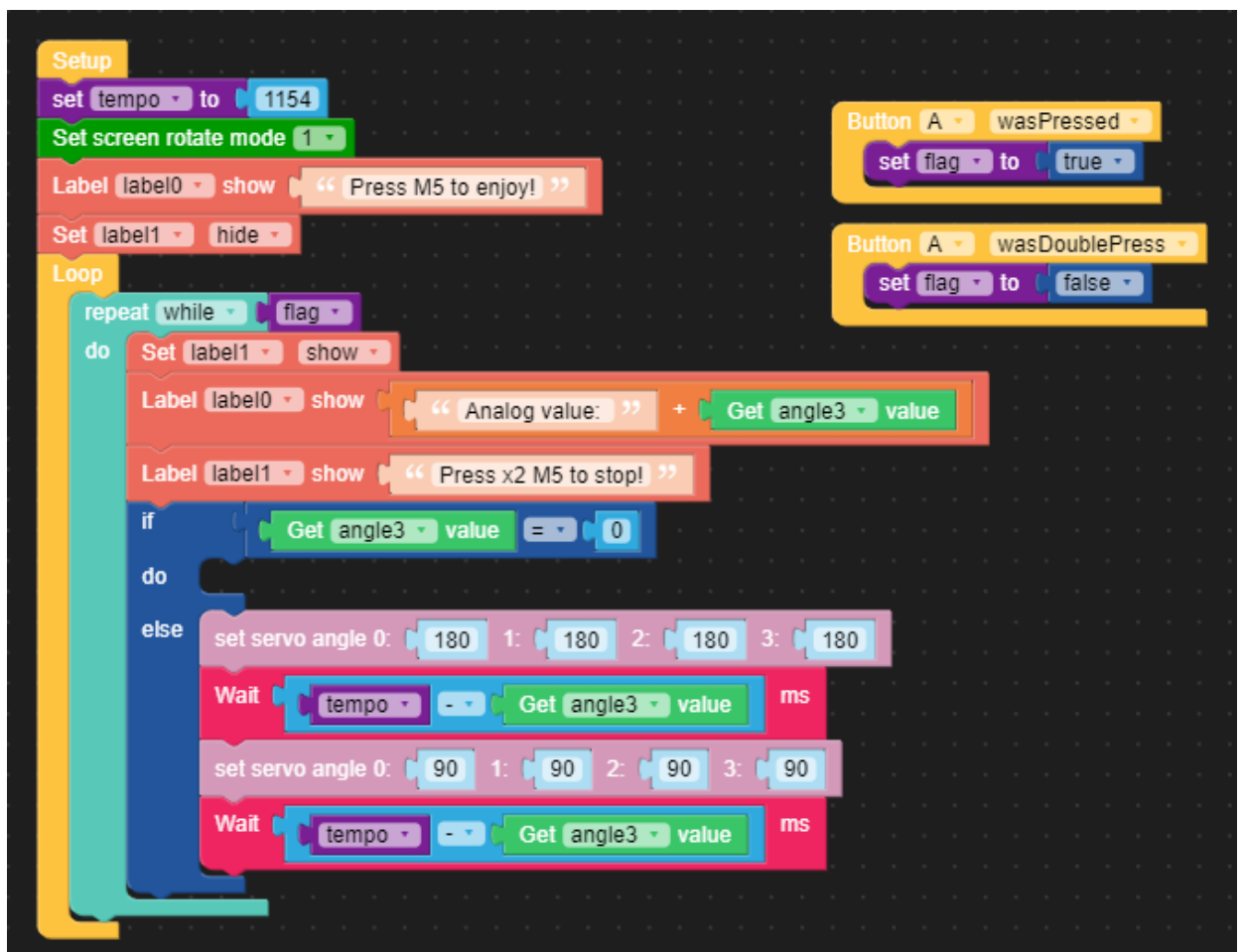


Figura 5.3.2 - Codice PuppyC - Trimmer

A questo punto, dopo aver scelto nell'IDE i dispositivi da usare, è stato testato il funzionamento del puppy con il potenziometro regolando la velocità. Il codice che è stato utilizzato è in figura 5.3.2.

Per quanto riguarda il funzionamento, innanzitutto è stata inizializzata la variabile “tempo” a 1154. Questo valore è stato ottenuto sommando il valore digitale massimo del potenziometro (che come è stato detto è 1024) al tempo minimo necessario affinché i servo del puppy compiano una rotazione completa da 180° a 90°, quindi in un tempo pari a 130 ms.

Successivamente la funzione “Set screen rotate mode 1” ruota lo schermo orizzontalmente e sono stati poi creati i campi di testo per inserire i vari messaggi che appariranno sull’LCD.

La variabile “flag” invece, è utile a far partire il ciclo while nel momento in cui viene premuto il tasto A e se quest’ultimo viene premuto due volte consecutive, il ciclo si ferma insieme al puppy. Per fare questo, sono stati utilizzati i blocchi “Button A was Pressed” e “Button A was Doublepress”.

All’interno del ciclo while invece, la funzione logica if-do-else serve a far spegnere il puppy quando il valore digitale del potenziometro, che si ricava attraverso la funzione “Get angle 3 value”, è 0.

Il blocco dedicato al puppy “set servo angle 0:” imposta l’angolo in cui deve ruotare il servo e la funzione wait rappresenta il tempo per spazzare un angolo di 90°. Ed è proprio quest’ultima che permette di controllare la velocità del dispositivo. Infatti man mano che viene ruotato l’albero del potenziometro, il tempo che passa da una commutazione all’altra, varia da 130 ms, che è il caso più veloce, a 1154 ms che è il caso più lento, grazie alla variabile “tempo” introdotta in precedenza. Il tutto viene inserito in un loop affinché venga ripetuto costantemente.

## 5.4 CONTROLLO REMOTO DI PUPPYC ATTRAVERSO L'ESPNow

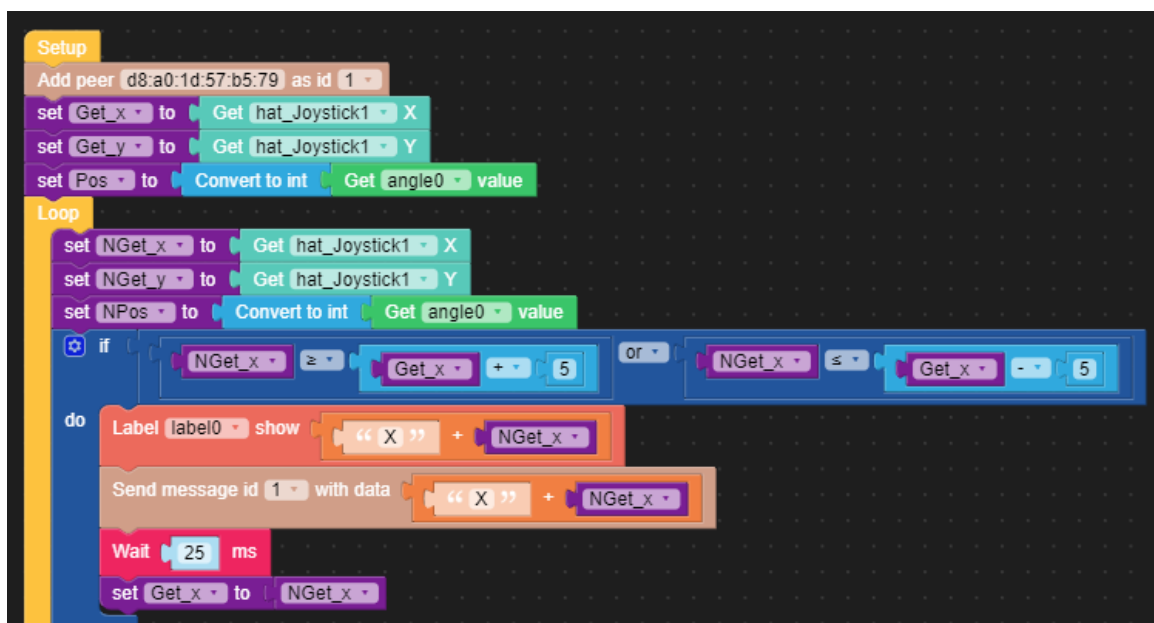
Affinché si possa ottenere un azionamento del puppy senza alcuni collegamenti circuitali e senza wifi, è stato opportuno far interagire il PuppyC con il potenziometro e il joystick con l'ausilio del protocollo di comunicazione ESPNow come anticipato nell'introduzione del capitolo.

Espnow è un protocollo di comunicazione peer to peer che consente a più dispositivi di comunicare tra loro senza utilizzare il Wi-Fi. Il protocollo è simile alla connettività wireless a 2,4 GHz a bassa potenza che viene spesso implementata ad esempio nei mouse wireless. Prima di tutto, è necessario l'accoppiamento tra i dispositivi prima della loro comunicazione, che avviene direttamente tramite i rispettivi indirizzi MAC. Dopo l'associazione, la connessione ptp è permanente.

In questo ultimo test, c'è da una parte l'M5 con il puppy e dall'altra un secondo M5 con il joystick hat e il potenziometro unit.

Il Joystick HAT è uno dei moduli HAT appositamente progettato per M5StickC. È stato utilizzato un microprocessore STM32F030F4 all'interno per implementare la comunicazione I2C con l'M5. Questo modulo dalla grandezza molto piccola, supporta il movimento angolare completo, la pressione della leva centrale ed emette dei dati angolari e dei segnali digitali.

Per quanto riguarda il codice, quest'ultimo è ovviamente diviso in due parti distinte: una contiene sostanzialmente come le istruzioni devono essere mandate dal joystick e dal potenziometro al puppy; l'altra contiene invece le istruzioni relative al puppy stesso, in cui vengono descritte tutte le azioni che deve fare quando riceve i dati. Verrà analizzato e commentato prima il codice denominato "Sender" presente nella figura 5.4.1.



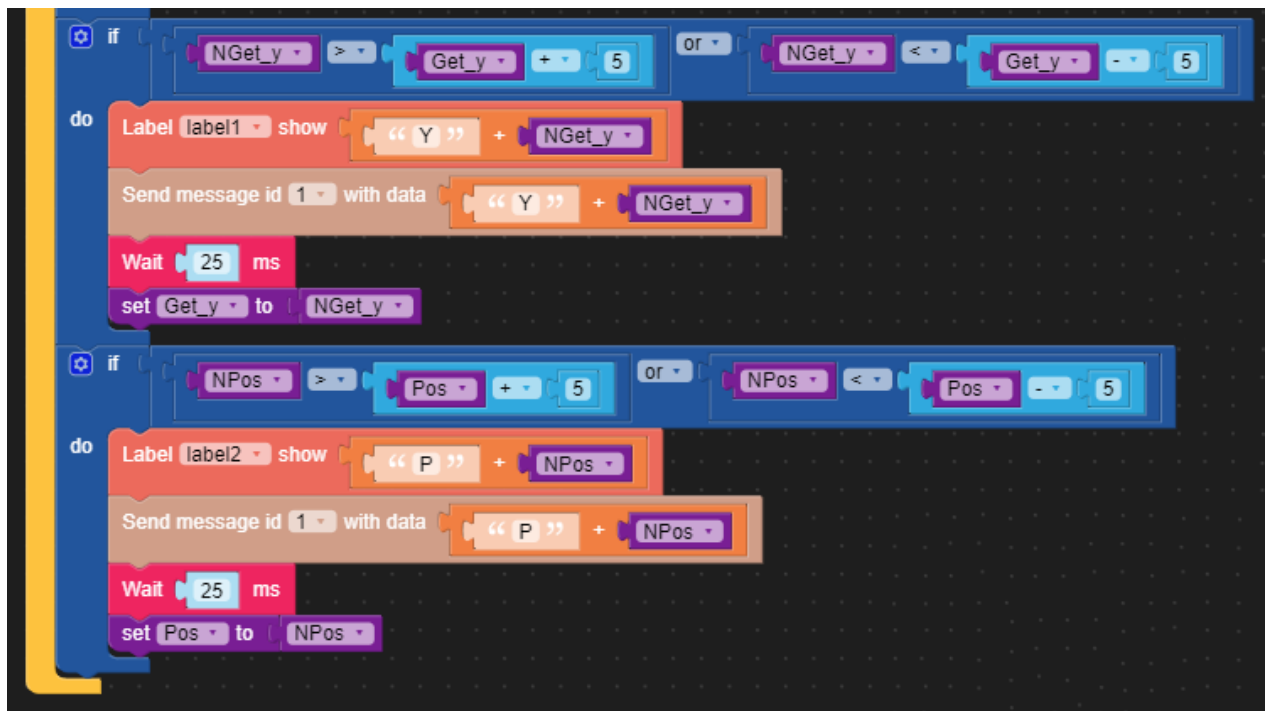


Figura 5.4.1 - Codice M5StickC con joystick e potenziometro

Come prima cosa, bisogna associare i due M5StickC per far sì che possano comunicare tra loro. Ciò è possibile attraverso il blocco “Add peer d8:a0:1d:57:b5:79 as id 1” che identifica l’M5 collegato al puppy, come dispositivo numero 1.

Sono state inizializzate le variabili globali “Get\_x”, “Get\_y” e “Pos” che corrispondono rispettivamente alla posizione del joystick rispetto l’asse x e y e al valore dell’angolo che imposta l’utente sul potenziometro. Quest’ultimo è stato convertito in intero per filtrare un minimo il dato.

All’interno del loop, sono state inizializzate altre tre “nuove” variabili che hanno gli stessi valori delle variabili precedenti.

Questo perché la funzione logica if-do, controlla che le variabili cambino di un valore pari a  $\pm 5$  rispetto al valore precedente e di conseguenza serviva iniziarle con nomi diversi ma contenuto uguale.

Infatti, per quanto riguarda il primo blocco logico ad esempio, se “NGet\_x” è maggiore di “Get\_x + 5” o minore di “Get\_x - 5”, allora l’istruzione va avanti e verrà mandato attraverso il blocco “Send message id 1 with data X+NGet\_x” solo e soltanto la stringa con il valore che ha uno scarto di 5. In questo modo viene filtrato ciò che viene mandato così da non avere dei dati inutili.

Il blocco “Label 0 show X+Nget\_x” mostra su schermo il valore che assume in quel momento l’albero del joystick lungo l’asse x.

Viene poi inserito una Wait da 25 ms per fare in modo che l’M5 che riceve riesca a distinguere i dati, altrimenti ne riceverebbe talmente tanti in pochissimo tempo, che non riuscirebbe a gestire.

Viene infine reinizializzata la variabile fuori dal loop per far continuare le istruzioni del blocco logico con un nuovo dato.

Le istruzioni relative alla variabile NGet\_y e Pos sono le medesime di quanto scritto finora.

Si può passare ora a commentare il codice denominato “Receive” relativo all’M5StickC che controllerà il puppy grazie ai dati che riceve attraverso l’ESPNow. È rappresentato nelle figure 5.4.2.1 e 5.4.2.2 in stile blockly come il precedente.

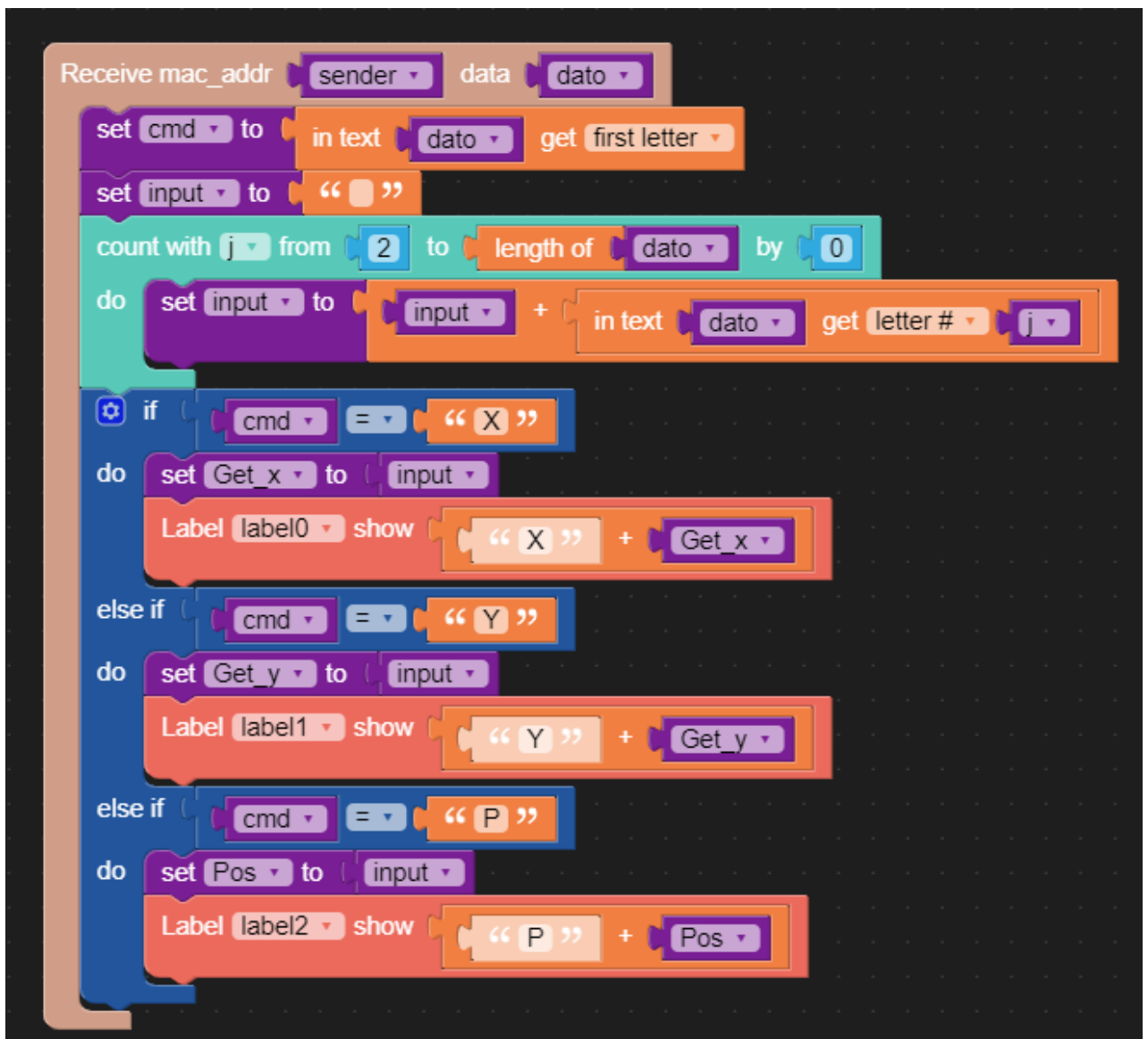


Figura 5.4.2.1 - Blocco receive relativo all’M5StickC con PuppyC



La funzione “Receive mac\_addr “sender” data “dato”” si appresta a ricevere dall’M5StickC che manda, i valori sotto forma di stringa chiamati “dato”.

Viene inizializzata la variabile “cmd” con il valore della prima lettera di ciò che gli arriva, attraverso il blocco “In text dato get first letter”.

Viene settata la variabile “input” ad una stringa vuota.

Viene avviato un ciclo while in cui una variabile contatore j, inizia a scorrere dalla seconda posizione fino a tutta la lunghezza della stringa “dato” che riceve, con un passo pari a 0.

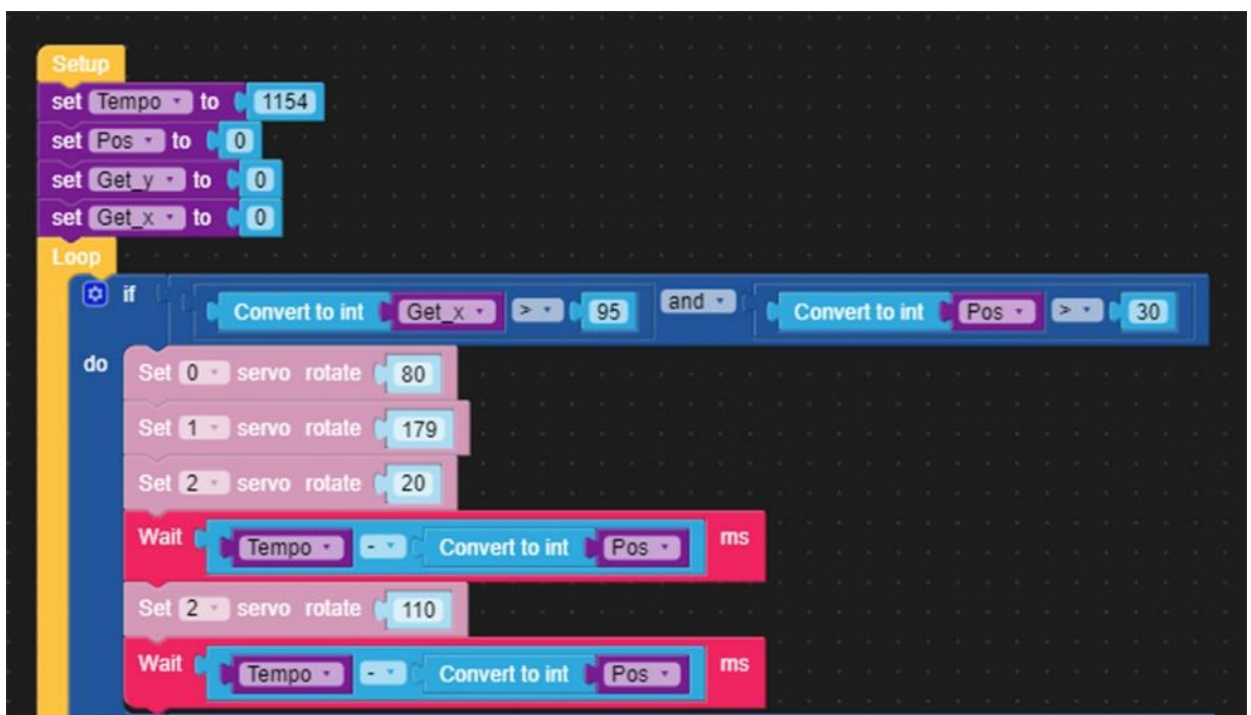
Nel frattempo, la variabile "input" ad ogni passaggio del ciclo, viene riempita con l’elemento j-esimo di "dato" a partire quindi dalla seconda lettera.

A questo punto, grazie al blocco logico if-do-else, è possibile dare il valore alla variabile che serve conoscere, a seconda della prima lettera della stringa.

Quindi se “cmd” è pari ad X, allora viene settato “Get\_x” con il valore di input che coincide con il valore digitale che è stato mandato dall’M5StickC attraverso il joystick.

Il blocco successivo mostra la variabile che arriva sull’M5 che è montato sul puppy. Questo procedimento si ripete anche per “Get\_y” e “Pos”.

Il setup vero e proprio del programma che sarà ciò che poi farà il puppy, si trova insieme al blocco receive ed è fatto in questo modo:



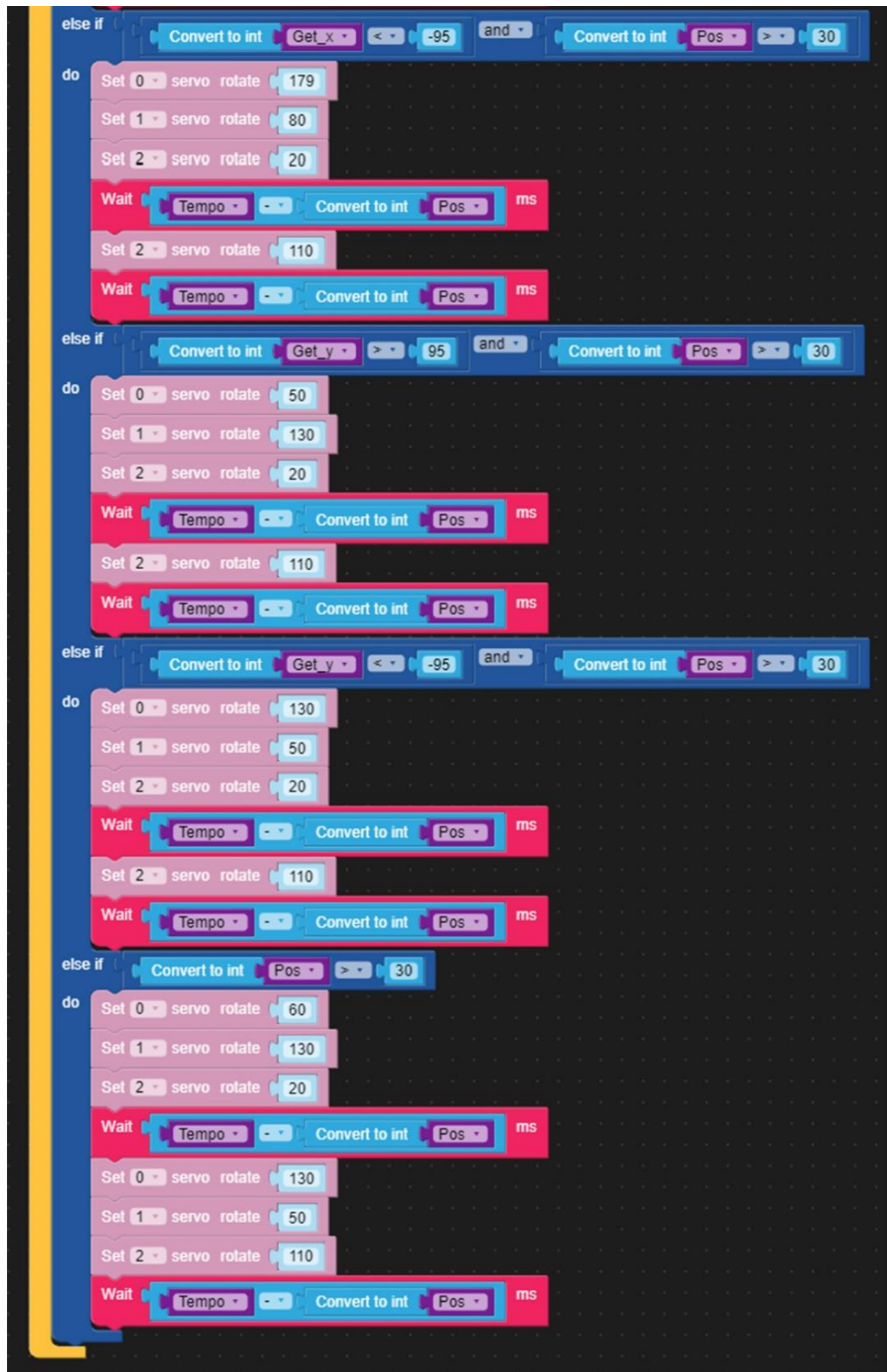


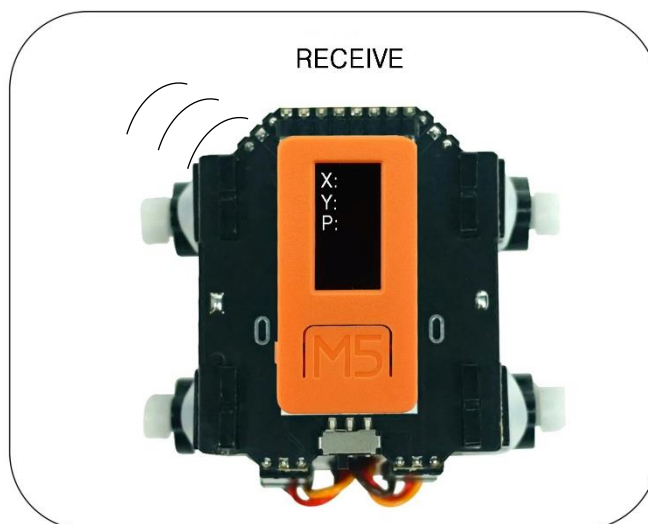
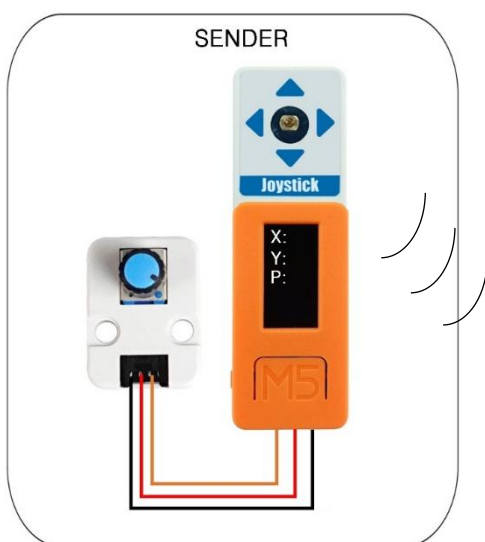
Figura 5.4.2.2 - Setup della receive relativo all'M5StickC con PuppyC

Innanzitutto vengono inizializzate le tre variabili “Pos” “Get\_x” e “Get\_y” a zero, che coincidono con quelle ricevute attraverso la receive.

Dato che si vuole controllare il puppy, è stata reintrodotta la variabile “tempo” per permettere il controllo della velocità. Il motivo del valore pari a 1154, è stato spiegato nel paragrafo 5.3.

All’interno del loop ci sono sostanzialmente i primi quattro blocchi logici che funzionano tutti allo stesso modo ma per direzioni del joystick diverse. Esaminando il primo blocco logico if-do-else if, viene imposta la condizione secondo la quale se il valore della posizione del joystick, quindi “Get\_x”, è maggiore di 95 (in un range da -130 a +130) e contemporaneamente il potenziometro ha un valore maggiore di 30, per eliminare così imprecisioni vicine lo zero, allora vengono settati i servo del puppy nel modo rappresentato dal codice. Vengono configurati solo tre servo perché il quarto non serve ai fini del progetto. Così facendo, vengono posizionati due motori in una posizione mentre il terzo continua a muoversi secondo la velocità fornita. L’ultimo blocco invece, regola solo la velocità di tutti e tre i servomotori, come è stato fatto durante le prime esperienze con il puppy.

Quindi sui display di entrambi gli M5StickC, durante il funzionamento di entrambi i codici, si potranno osservare i valori che assumono le ascisse e le ordinate del joystick (X e Y) e il valore equivalente del potenziometro (P). Inoltre alla luce di quanto detto finora, si può riassumere il tutto con due rappresentazioni, una della Sender e l’altra della Receive in questo modo:



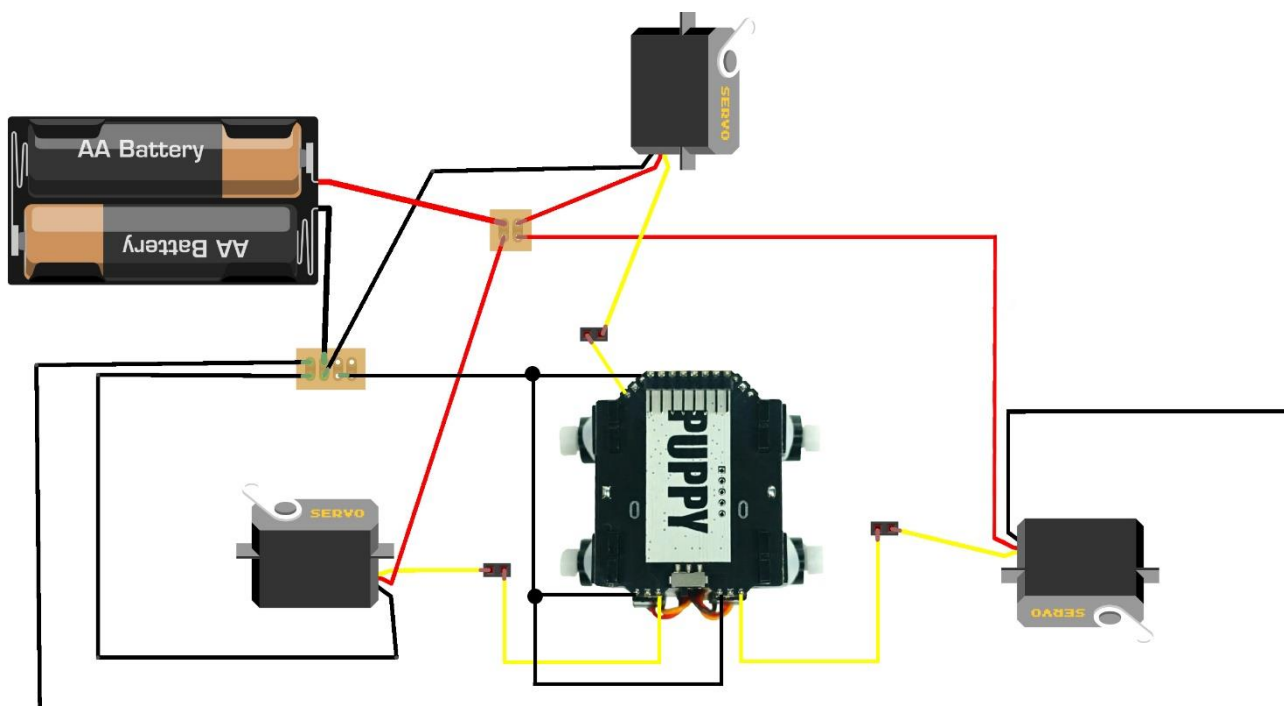
## 5.5 ADATTAMENTO DELLO SCHEMA DI NAVIGAZIONE E GUIDA A BORRACCIOLO

È utile a questo punto spiegare come tutto ciò di cui si è parlato in questo capitolo, è stato adattato al prototipo modellato in precedenza.

Per quanto riguarda lo schema di navigazione, quest'ultimo corrisponde dal punto di vista meccanico, proprio al puppy che è stato controllato con l'ESPNow. All'interno del progetto, rappresenta proprio l'attuatore, quindi ciò che permetterà al pesce robot la navigazione in acqua. Infatti si è pensato di inserire all'interno del corpo della borraccia, l'alimentazione erogata dalla batteria e il microcontrollore che rappresenta l'intelligenza del robot, collegato al puppy. Successivamente, dato che il puppy contiene dei servomotori molto piccoli, sono stati collegati dei savox (che sono stati nominati nei capitoli precedenti relativi alla modellazione) più potenti, così da permettere al prototipo di nuotare velocemente. Quindi ora è ben chiaro che il codice relativo alla "Receive", è il corrispondente software del sistema di navigazione. Infatti riceve i dati e muove le pinne che compongono il pesce in base a come l'operatore comanda con il joystick e il potenziometro. È proprio per questo motivo che all'interno dei codici relativi alla "Receive", sono stati settati solo tre servo, dato che le pinne in movimento sono le due mediane e la pinna caudale. A tal proposito, conviene spiegare il movimento corrispondente che il pesce fa quando riceve determinati comandi. Se il pesce riceve un input del joystick lungo l'asse y verso l'alto o verso il basso, le pinne mediane si posizioneranno con la parte posteriore rispettivamente verso il basso o verso l'alto così da salire in superficie o andare in profondità, tutto questo mentre la pinna caudale è in movimento continuo per permettere la spinta posteriore. Nel caso in cui invece viene captato un segnale del joystick lungo l'asse x verso sinistra o verso destra, le pinne mediane si posizioneranno in modo che rispettivamente la destra e la sinistra "taglino" l'acqua, mentre ovviamente la pinna posteriore continua a spingere il prototipo in avanti. Nel caso in cui viene solo accelerato il pesce, le pinne si muovono in modo da far nuotare il robot come un pesce reale.

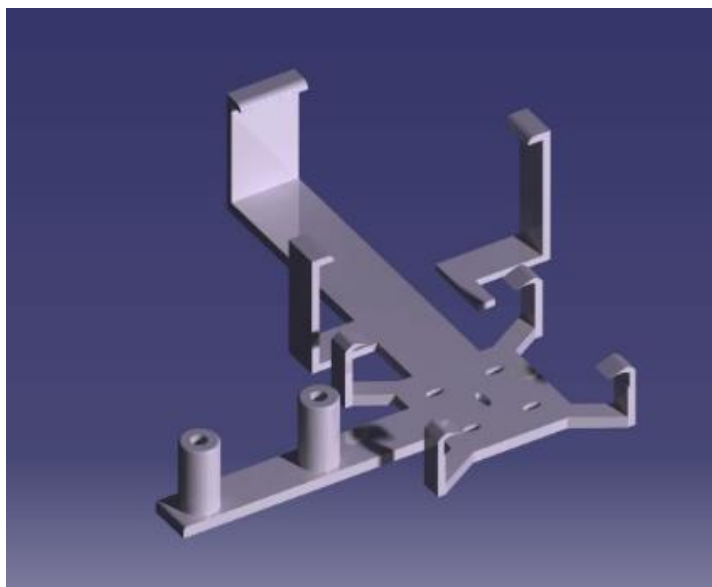
Di seguito un'immagine che mostra in modo chiaro la componente circuitale all'interno del corpo del robot, cioè nella borraccia. Le masse, così come le alimentazioni, sono state tutte collegate insieme. I segnali PWM (in giallo) sono stati collegati ognuno dall'uscita del puppy al servo corrispondente nel sistema del pesce robot. I collegamenti sono stati fatti con l'ausilio di un saldatore a stagno e scotch isolante per evitare corto circuiti tra le varie componenti.

Inoltre nel momento in cui verrà provato in acqua, verrà impermeabilizzato il tutto con l'ausilio di resina e colla a caldo.



*Figura 5.5.1 - Circuito interno del prototipo*

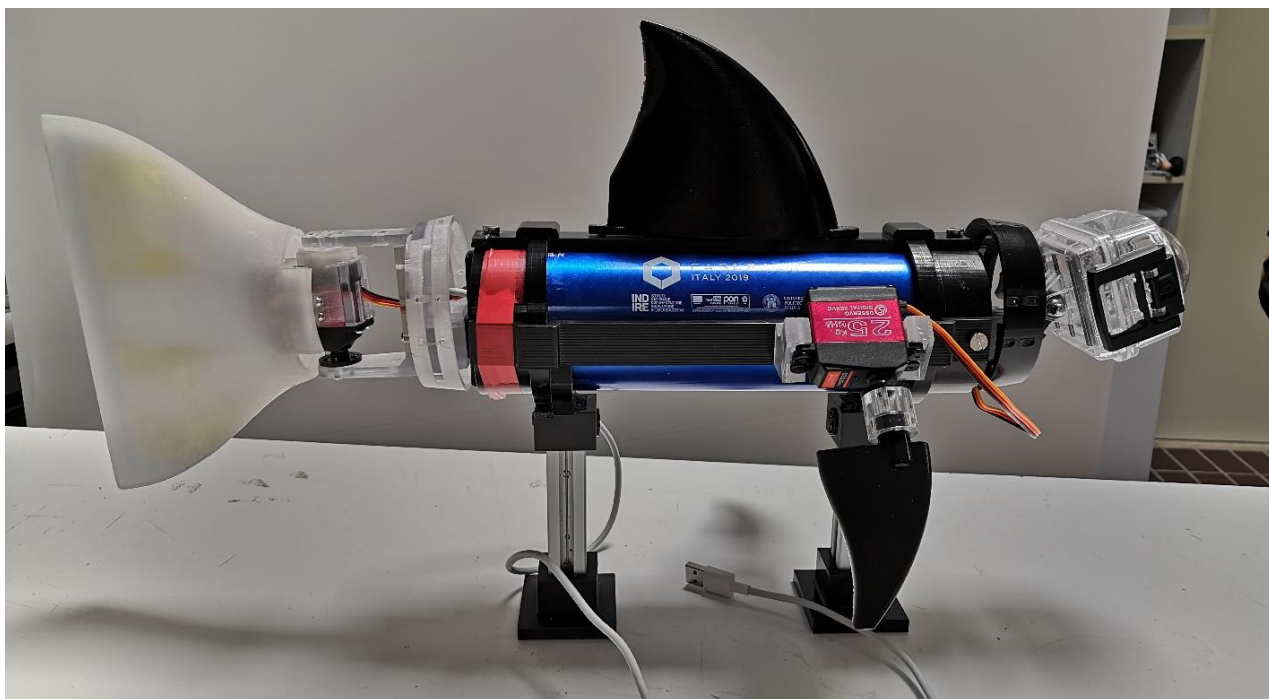
Per quanto riguarda il sistema di guida, a questo punto non è nient'altro che la combinazione M5StickC-Joystick-Potenzimetro. La batteria utilizzata, nel caso di un lungo utilizzo, può essere un powerbank con alimentazione da 5V, dato che l'M5StickC ha una batteria piuttosto piccola. Quindi chi ha in mano il tutto, può decidere la velocità e le direzioni che il pesce robot deve prendere. Infatti, dal punto di vista software, il corrispondente del sistema di guida è la "Sender" che manda in continuazione gli input all'M5StickC che si trova sul puppy. È stata creata inoltre una struttura per tenere insieme i tre dispositivi durante il controllo.



*Figura 5.5.2 - Struttura joystick-M5-potenzimetro*



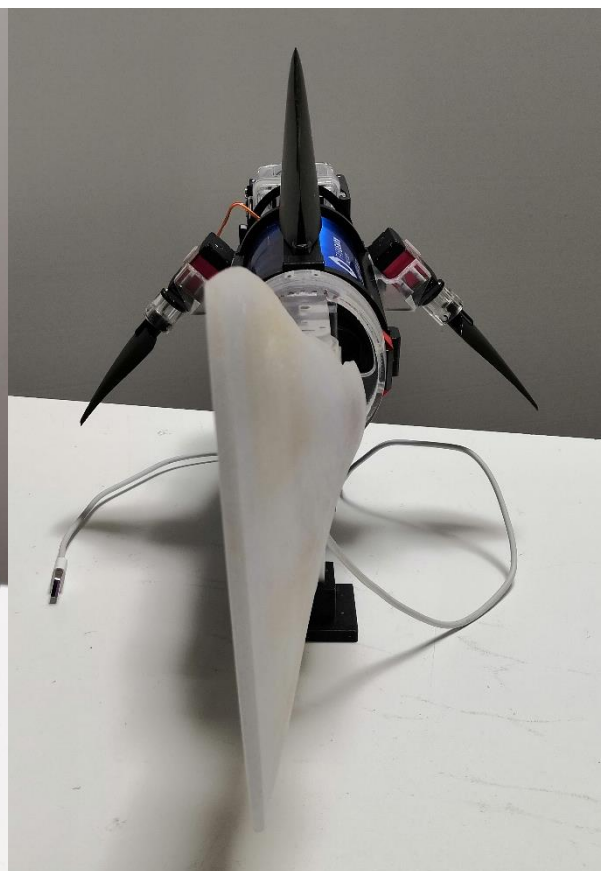
Di seguito infine, alcune foto del prototipo completo montato in laboratorio.



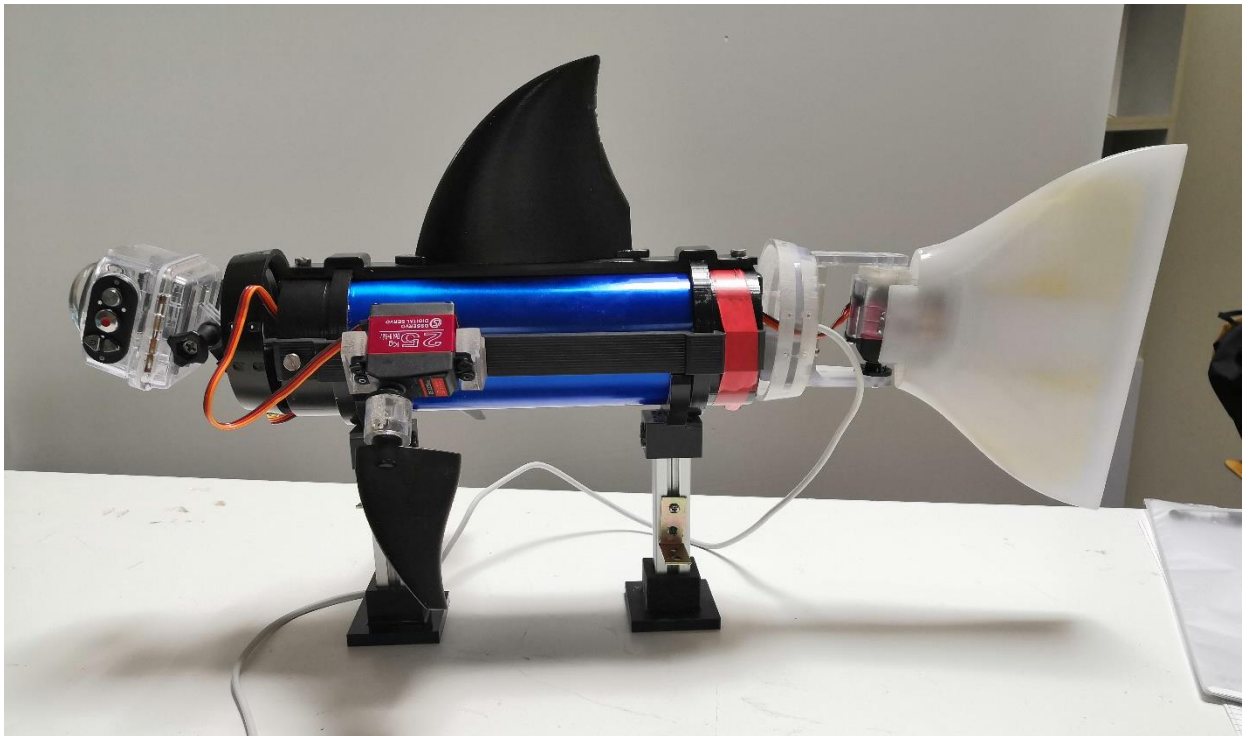
*Figura 5.5.3 - Vista laterale destra Borracciolo 1.0*



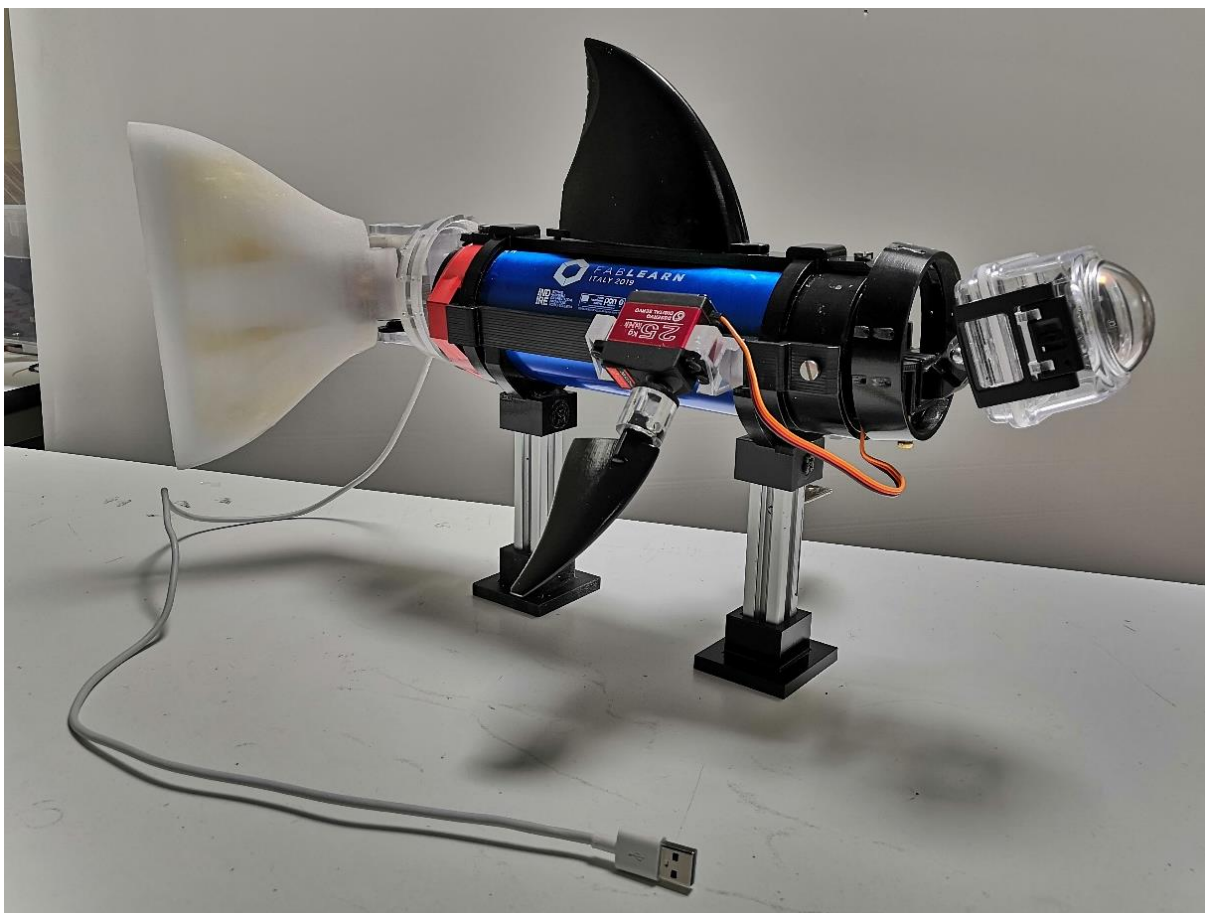
*Figura 5.5.4 - Vista frontale Borracciolo 1.0*



*Figura 5.5.5 - Vista posteriore Borracciolo 1.0*



*Figura 5.5.6 - Vista laterale sinistra Borracciolo 1.0*



*Figura 5.5.7 - Vista isometrica Borracciolo 1.0*

## CONCLUSIONI

Il progetto di design e controllo del prototipo è stato portato a termine seguendo il più possibile le nozioni apprese durante il percorso di studi all'interno dell'UNIVPM.

Le modifiche apportate al modello con l'uso del software Catia, sono state svolte in maniera efficace ed offrono una buona base di partenza per ogni possibile proposta di miglioramento futura.

Nell'analisi agli elementi finiti sono stati posti dei vincoli e successivamente è stata svolta una simulazione che ha permesso non solo di comprendere al meglio il comportamento del materiale utilizzato, ma anche di proporre, delle soluzioni geometriche semplici ed efficaci per contenere gli sforzi presenti nella struttura.

Per ciò che concerne il controllo, sono stati scelti degli strumenti di programmazione che sono adatti allo scopo del presente elaborato, cioè alla robotica educativa. I bambini infatti si troveranno alle prese con la creazione di programmi in modo semplice, intuitivo e divertente. Inoltre studiare e applicare robotica educativa e coding non è importante soltanto per imparare a costruire e programmare i robot, ma anche per imparare un metodo di ragionamento e sperimentazione. Tutto ciò promuove le attitudini creative degli studenti, nonché (Ippoliti s.d.) la loro capacità di comunicazione, cooperazione e lavoro di gruppo. Si tratta quindi di indirizzare i ragazzi ad un nuovo metodo di studio basato sui concetti di problem solving e sul learn by doing.

Purtroppo non è stato possibile per cause di forza maggiori (Covid-19) testare il prototipo in acqua. In ogni caso però è stato possibile fare dei test in classe in cui è stato azionato il robot. Sfortunatamente si è notato che un uso prolungato del prototipo, porta ad un riempimento della memoria flash, dato che è molto piccola (4 Mb). Questo inconveniente si può risolvere solo spegnendo l'M5StickC presente all'interno della borraccia. Ciò non permette di fermare completamente le pinne del prototipo, ma solo di rallentare al massimo l'azionamento, e in ogni caso si ha un ritardo generale dell'intero sistema.

A tal proposito, negli sviluppi futuri si cercherà di studiare e risolvere questo problema. Inoltre si tenderà a rendere il prototipo ancora più leggero e funzionale, oltre a risolvere eventuali problematiche, attraverso uno studio dinamico, che si potrebbero riscontrare durante i test in acqua.

Per finire, ci si può quindi ritenere comunque soddisfatti nell'aver portato a termine questa prima fase di studio con risultati soddisfacenti e aprendo la strada a chiunque avrà, negli anni che verranno, la possibilità di studiare soluzioni volte a migliorare la dinamica del prototipo.



## BIBLIOGRAFIA

David Scaradozzi, Laura Sorbi, Anna Pedale, Mariantonietta Valzano, Cinzia Vergine. «Teaching robotics at the primary school: an innovative approach.» *ScienceDirect*, 2014: 9.

Ippoliti, Gianluca. «Tecnologie per l'automazione e la robotica.» s.d.

Paola Pazzaglia, David Scaradozzi. «Digital Competencies, Coding, Educational Robotics, STEM, Primary School, Earthquake.» s.d.

Scaradozzi, D., Cesaretti, L., Screpanti, L., Costa, D., Zingaretti, S., & Valzano, M. *Innovative tools for teaching marine robotics, iot and control strategies since the primary school. In Smart Learning with Educational Robotics*. Cham: Springer, 2019.

Vario, Maddalena. «Come AI e robotica cambieranno le nostre vite.» *Garr news*, Dicembre 2018: 4-6.

Vignocchi, Daniela. *Progettare con catia v5*. Seconda. Athena, 2005.

## SITOGRAFIA

<https://www.robopisces.eu/>

<https://www.stratasys.com/>

<https://www.stampanti3d.it/>

<https://www.cad3d.it/>

<https://www.wikipedia.it/>

<https://www.maffucci.it/tag/m5stickc/>

<https://www.campustore.it/robotica-e-coding.html/>

## RINGRAZIAMENTI

*Giunto alla fine di questo percorso universitario, desidero ringraziare tutti coloro che mi hanno accompagnato e sostenuto fino a questo importante traguardo.*

*Per prima, vorrei ringraziare di cuore la mia famiglia, senza la quale non sarei mai riuscito ad arrivare a questo punto, grazie per avermi supportato con tanti sacrifici, pane e senza pane, nei momenti più difficili e per avermi sempre sostenuto qualunque sia stata la strada da me scelta. A voi devo tutto.*

*Un ringraziamento speciale va ai miei tre coinquilini e amici, con i quali ho condiviso momenti di gioia e tristezza e mi hanno accompagnato durante questi anni con suggerimenti, critiche e tutto ciò che mi ha reso quello che sono adesso.*

*Ringrazio con affetto la mia ragazza e amica, che mi ha sempre assecondato ed è stata sempre dalla mia parte, rappresentando un'ancora fondamentale ogni qualvolta ne avessi bisogno.*

*Vorrei ringraziare immensamente tutti i miei amici e colleghi che sono stati presenti in questi ultimi anni, con i quali le giornate di studio terminavano con momenti allegri e lieti. Non avrei mai pensato di conoscere persone così speciali come lo siete voi.*

*Un ringraziamento speciale va a tutti i miei amici che sono stati lontani fisicamente ma vicini col cuore, senza farmi notare quella che è la distanza che ci separa ormai da vari anni.*

*Per ultimo, ma non meno importante, vorrei ringraziare particolarmente il Professor David Scaradozzi, relatore di questa tesi, non solo per l'attenzione che mi ha dedicato durante tutto il periodo di tirocinio e di stesura della tesi con critiche e suggerimenti, ma anche per la passione che dimostra nel suo lavoro e che soprattutto riesce a trasmettere durante tutta quest'ultima fase del percorso universitario. Grazie anche alla sua simpatia, professionalità e cordialità, ho potuto apprendere con relativa leggerezza tutte le nozioni necessarie allo sviluppo del robot.*

*Fernando*