

POLITECNICO DI TORINO  
Dipartimento di Ingegneria Meccanica e Aerospaziale

Tesi di Laurea Magistrale

Sviluppo di un sistema di rientro con  
controllo di traiettoria per palloni  
sonda



Candidato	Victor Miherea
Tutor Accademico	Prof. Giorgio Guglieri
Tutor Accademico	Prof. Paolo Maggiore
Tutor Aziendale	Dott. Ivano Musso

Settembre 2018





# Premessa

Le HAP (*High Altitude Platform*) sono piattaforme operanti ad alta quota tra i 20 e i 50 km, con una posizione fissa rispetto alla Terra. Questi aeromobili, in genere dirigibili, palloni o velivoli senza equipaggio, hanno l'obiettivo di comporre una rete di telecomunicazioni o di eseguire un telerilevamento, per scopi civili e militari. Queste stazioni, dal punto di vista dei costi, possono diventare una valida alternativa ai satelliti: esistono molti progetti in fase di sviluppo. Il recupero completo del modulo di rientro a Terra abbasserebbe ancora di più i costi. La tesi di laurea magistrale, qui presentata, svolta in collaborazione con *ALTEC Space*, tratta dello sviluppo in scala di un prototipo di sonda, chiamato *Cube-HAP*: un sistema di rientro con controllo di traiettoria, che utilizza un pallone stratosferico, un parafoil e un sistema di ventole per la sua movimentazione. Lo svolgimento del lavoro, avvenuto da febbraio ad agosto del 2018, include la progettazione di un sistema di pilotaggio remoto e controllato, con il relativo studio del hardware presente a bordo, la sua integrazione sulle configurazioni complete, la loro costruzione e la convalida dei test di volo preliminari, concentrandosi sulla parte di pilotaggio del parafoil nella fase finale di rientro.



# Riconoscimenti

Ringrazio *Altec Space* per il supporto tecnico e l'aiuto offerti durante lo svolgimento del lavoro. Un grazie speciale va al prof. Giorgio Guglieri, al dott. Ivano Musso e al Sig. Paolo Navone per la loro supervisione, il loro aiuto, la loro disponibilità e i loro consigli tecnici e non.

Ringrazio la mia famiglia.

Ringrazio i miei Amici.



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Motivazioni e obiettivi . . . . .	1
1.2	Panoramica sui palloni stratosferici . . . . .	2
1.2.1	Vantaggi nell'utilizzo dei palloni sonda . . . . .	2
1.2.2	Regolamento . . . . .	4
1.2.3	Profilo di missione . . . . .	5
1.2.4	Applicazioni delle HAPS . . . . .	6
<b>2</b>	<b>Progettazione</b>	<b>9</b>
2.1	Funzionamento del sistema . . . . .	9
2.2	Configurazione preliminare . . . . .	10
2.2.1	Requisiti . . . . .	10
2.2.1.1	Pesi e ingombri . . . . .	10
2.2.2	Elettronica di bordo . . . . .	10
2.2.3	Costo dell'elettronica di bordo . . . . .	16
2.3	Componenti Strutturali . . . . .	16
2.4	Disegno CAD . . . . .	19
2.4.1	Configurazione 1 . . . . .	19
2.4.1.1	Contenitore Parafoil . . . . .	19
2.4.1.2	Anello superiore . . . . .	19
2.4.1.3	Contenitore Raspberry . . . . .	19
2.4.1.4	Contenitore Batteria . . . . .	21
2.4.2	Configurazione 2 . . . . .	22
2.4.2.1	Supporto Superiore . . . . .	22
2.4.2.2	Supporti dei servi . . . . .	22
2.4.2.3	Supporto inferiore . . . . .	22
2.5	Stampa 3D . . . . .	24
<b>3</b>	<b>Integrazione</b>	<b>27</b>
3.1	Integrazione dell'elettronica di bordo . . . . .	27
3.1.1	Schema elettrico . . . . .	27
3.1.2	Collegamento tra i dispositivi . . . . .	27
3.1.2.1	GPS e IMU collegati al Raspberry Pi Zero . . . . .	29
3.1.2.2	Servomeccanismi ed ESC collegati al Maestro . . . . .	30
3.1.2.3	Collegamento: batteria, ESC e motori, Maestro e Raspberry . . . . .	33
3.1.3	Durata della batteria . . . . .	33
3.2	Integrazione della struttura con l'elettronica di bordo . . . . .	34

3.2.1	Configurazione 1 . . . . .	35
3.2.1.1	Protezione aggiuntiva per la Configurazione 1 . . . . .	35
3.2.2	Configurazione 2 . . . . .	38
3.2.2.1	Protezione aggiuntiva per la Configurazione 2 . . . . .	43
<b>4</b>	<b>Impostazione dei software</b>	<b>45</b>
4.1	Raspberry Pi Zero W . . . . .	45
4.2	Attivazione del GPIO . . . . .	49
4.3	Attivazione dell'unità GNSS . . . . .	51
4.3.0.1	Test sul funzionamento del gpsd . . . . .	52
4.4	Attivazione della piattaforma inerziale . . . . .	52
4.4.1	minimu9-ahrs . . . . .	53
4.4.2	ahrs-visualizer . . . . .	53
4.4.3	Introduzione al filtro di Madgwick . . . . .	55
4.5	Maestro . . . . .	55
4.5.1	Configurazione del Maestro e inizializzazione dei servi . . . . .	56
4.5.1.1	Status (Maestro) . . . . .	56
4.5.1.2	Channel Settings (Maestro) . . . . .	57
4.5.2	Connessione del Maestro con il Raspberry Pi . . . . .	59
4.6	Software per il controllo del sistema . . . . .	59
4.6.1	Controllo autonomo della direzione del volo . . . . .	59
4.6.2	Controllo manuale della traiettoria . . . . .	60
4.7	Implementazione complessiva dei software . . . . .	62
<b>5</b>	<b>Test e Convalida</b>	<b>67</b>
5.1	Test sulla durata della batteria . . . . .	67
5.2	Test sulla compatibilità elettromagnetica dell'IMU . . . . .	69
5.3	Test sul corretto funzionamento dell'unità GPS . . . . .	69
5.4	Preparazione al test preliminare di volo . . . . .	69
5.4.1	Utilizzo della sonda di rientro . . . . .	69
5.4.2	Calibrazione giroscopio . . . . .	73
5.4.3	Calibrazione magnetometro . . . . .	73
5.4.3.1	Calibrazione iniziale da minimu9-ahrs . . . . .	74
5.4.3.2	Calibrazione manuale . . . . .	74
5.4.4	Aggiornamento Offset e Compilazione . . . . .	74
5.5	Test preliminari di volo . . . . .	76
5.5.1	Test di volo 1: interno . . . . .	77
5.5.2	Test di volo 2: esterno . . . . .	77
5.5.2.1	Post-Processing dei dati del Test 2 . . . . .	77
<b>6</b>	<b>Conclusioni</b>	<b>83</b>
6.1	Raggiungimento degli obiettivi . . . . .	83
6.2	Criticità riscontrate . . . . .	83
6.3	Sviluppi futuri . . . . .	84
6.3.1	Sistema di espulsione del Parafoil . . . . .	84
6.3.2	Studio sulla spinta delle ventole . . . . .	84
6.3.3	Test aggiuntivi all'esterno . . . . .	85
6.3.4	Ulteriore sviluppo del software di controllo . . . . .	85
6.3.5	Ottimizzazione del peso . . . . .	85

Bibliografia . . . . .	85
------------------------	----



# Elenco delle figure

1.1	Pallone sonda . . . . .	3
1.2	Simulazione traiettoria . . . . .	5
1.3	Siti di lancio e sistema globale di osservazione . . . . .	7
1.4	Stratobus™ della Thales Alenia Space . . . . .	8
2.1	Il Parafoil utilizzato . . . . .	11
2.2	Raspberry Pi Zero W . . . . .	12
2.3	Accumulatore LiPo 2S 1000mAh . . . . .	13
2.4	Regolatore di tensione step-down . . . . .	14
2.5	Unità GPS e IMU utilizzati . . . . .	14
2.6	Maestro . . . . .	15
2.7	Regolatore elettronico di velocità con le ventole intubate . . . . .	16
2.8	Servomeccanismi utilizzati . . . . .	17
2.9	Contenitore del Parafoil . . . . .	19
2.10	Anello superiore della prima configurazione . . . . .	20
2.11	Controventatura . . . . .	20
2.12	Contenitore Raspberry Pi Zero . . . . .	21
2.13	Contenitore Batteria . . . . .	21
2.14	Supporto Superiore della seconda configurazione . . . . .	22
2.15	Supporti ausiliari seconda configurazione . . . . .	23
2.16	Supporto inferiore seconda configurazione . . . . .	23
2.17	Software Ultimaker Cura . . . . .	24
2.18	Pezzi della seconda configurazione stampati in 3D . . . . .	26
2.19	Supporto superiore della seconda configurazione stampato . . . . .	26
3.1	Schema elettrico generale . . . . .	28
3.2	Cavi utilizzati . . . . .	30
3.3	Collegamento tra Raspberry, unità GPS e IMU . . . . .	31
3.4	Collegamento tra Maestro, servomeccanismi ed ESC . . . . .	32
3.5	Collegamento tra batteria, ESC, motori, Maestro e Raspberry . . . . .	33
3.6	Integrazione della struttura con i sensori (prima configurazione) . . . . .	36
3.7	Integrazione della struttura con la batteria (prima configurazione) . . . . .	37
3.8	Integrazione della struttura con il Raspberry (prima configurazione) . . . . .	37
3.9	Struttura irrobustita della prima configurazione . . . . .	38
3.10	Aerofreni . . . . .	39
3.11	Prima configurazione assemblata . . . . .	40
3.12	Integrazione della parte superiore (seconda configurazione) . . . . .	41
3.13	Integrazione della parte inferiore (seconda configurazione) . . . . .	42
3.14	Seconda configurazione assemblata . . . . .	44

4.1	Installazione Raspbian . . . . .	46
4.2	Attivazione delle periferiche . . . . .	47
4.3	Schema connessione in remoto . . . . .	48
4.4	Schermata iniziale di Putty . . . . .	49
4.5	Schermata dei pin del GPIO . . . . .	50
4.6	Test sul corretto funzionamento dell'unità GNSS . . . . .	53
4.7	Dati in output di minimu9-ahrs: angoli di Eulero . . . . .	54
4.8	Applicazione del ahrs-visualizer . . . . .	54
4.9	Schermata iniziale di Pololu Maestro Control Center . . . . .	56
4.10	Impostazione dei canali per le condizioni iniziali . . . . .	58
4.11	Logica di PWM (Pulse Width Modulation) . . . . .	58
4.12	Movimentazione delle leve del Parafoil . . . . .	61
4.13	Movimentazione delle ventole intubate . . . . .	63
4.14	Schema funzionamento logico dei software . . . . .	65
5.1	Cicalino . . . . .	68
5.2	Andamento del voltaggio della batteria . . . . .	68
5.3	Effetto campo magnetico sul magnetometro . . . . .	70
5.4	Variazione degli angoli di assetto al variare della distanza del campo magnetico disturbante . . . . .	71
5.5	Percorso tracciato secondo le due unità GPS del cellulare . . . . .	72
5.6	Calibrazione del giroscopio . . . . .	73
5.7	Lo slittamento del campo magnetico dovuto all'effetto hard iron . . . . .	75
5.8	Lo slittamento del campo magnetico dovuto all'effetto soft iron . . . . .	75
5.9	Test 1 - Indoor . . . . .	77
5.10	Test 2 - Outdoor . . . . .	78
5.11	Traiettoria nelle tre dimensioni e in funzione del tempo - test outdoor . . . . .	79
5.12	Angoli di imbardata e beccheggio - test outdoor . . . . .	80
5.13	Accelerazioni nel tempo - test outdoor . . . . .	81
5.14	Velocità giroscopiche nel tempo - test outdoor . . . . .	81
5.15	Campi magnetici nel tempo - test outdoor . . . . .	81

# Elenco delle tabelle

2.1	Pesi e ingombri dei componenti elettronici . . . . .	11
2.2	Costo dell'elettronica di bordo . . . . .	18
2.3	Pesi e ingombri delle strutture . . . . .	25
3.1	Correnti assorbite e voltaggi nominali . . . . .	29
5.1	Confronto della durata della batteria teorica e pratica . . . . .	68



# 1 | Introduzione

Nella prima parte di questo capitolo vengono descritti le motivazioni e gli obiettivi preposti per lo svolgimento di questo lavoro, mentre nella seconda parte viene data una breve panoramica sul pallone stratosferico, sul profilo di missione e in quali applicazioni può essere impiegato.

## 1.1 Motivazioni e obiettivi

L'interessamento per questo argomento è legato alla ricerca di un'alternativa *low-cost* alle già approvate tecnologie dei satelliti artificiali, geostazionari e non, o alle tecnologie dei razzi-sonda, che vengono impiegate nel campo della comunicazione, della ricerca meteorologica, delle scienze legate all'osservazione astrale e quella terrestre e dei più svariati ambiti di ricerca. L'utilizzo dei palloni stratosferici può essere una valida alternativa, poiché ormai si è consolidata come una tecnologia affidabile e a basso impatto economico. Un passo ulteriore nel miglioramento di tale tecnologia è creare un sistema di guida autonoma, che permetta alla sonda, una volta sganciata dal pallone, di rientrare in una zona di atterraggio sicura, minimizzando così il rischio di perdere il payload, col vantaggio di abbassare ancora di più i costi. In altre parole il contenitore del *payload* diventa anche un sistema di rientro controllato con il parafoil.

Aziende come *X* del gruppo *Google Inc.* hanno avviato progetti di questo tipo, come ad esempio ***Loon Project*** [26] che ha l'obiettivo di portare la connessione internet in zone rurali, comunità remote o zone che hanno subito cataclismi naturali a livello globale, attraverso l'utilizzo di una flotta di palloni stratosferici. Anche *Facebook* con il loro progetto ***Aquila*** [27] ha cercato di utilizzare una flotta di APR, ad ala fissa, con lo stesso scopo, ma recentemente (giugno 2018) è stata annunciata da parte di Facebook la rinuncia a tale progetto [28].

Questi due esempi ci fanno capire quale potenziale mercato ci sia dietro a questa tecnologia non solo nel campo delle osservazioni scientifiche, ma anche nel campo della comunicazione di massa.

Infine, motivazioni di tipo più pratico, come la possibilità di lavorare su un progetto, di cui verrà creato un modello fisico, sono state alla base della scelta di questo tema.

Per quello che riguarda gli obiettivi, si cerca di sviluppare un prototipo di una sonda di rientro, controllata attraverso la movimentazione delle linee di comando di un parafoil e con l'ausilio delle ventole intubate. Nel rispetto dei requisiti iniziali, si cercherà di progettare e sviluppare delle configurazioni in grado di svolgere questo compito. Quindi ci si concentra sulla parte finale, quella legata al rientro controllato

a Terra. Infine, si cercherà di convalidare la bontà del progetto attraverso test di volo preliminari.

## 1.2 Panoramica sui palloni stratosferici

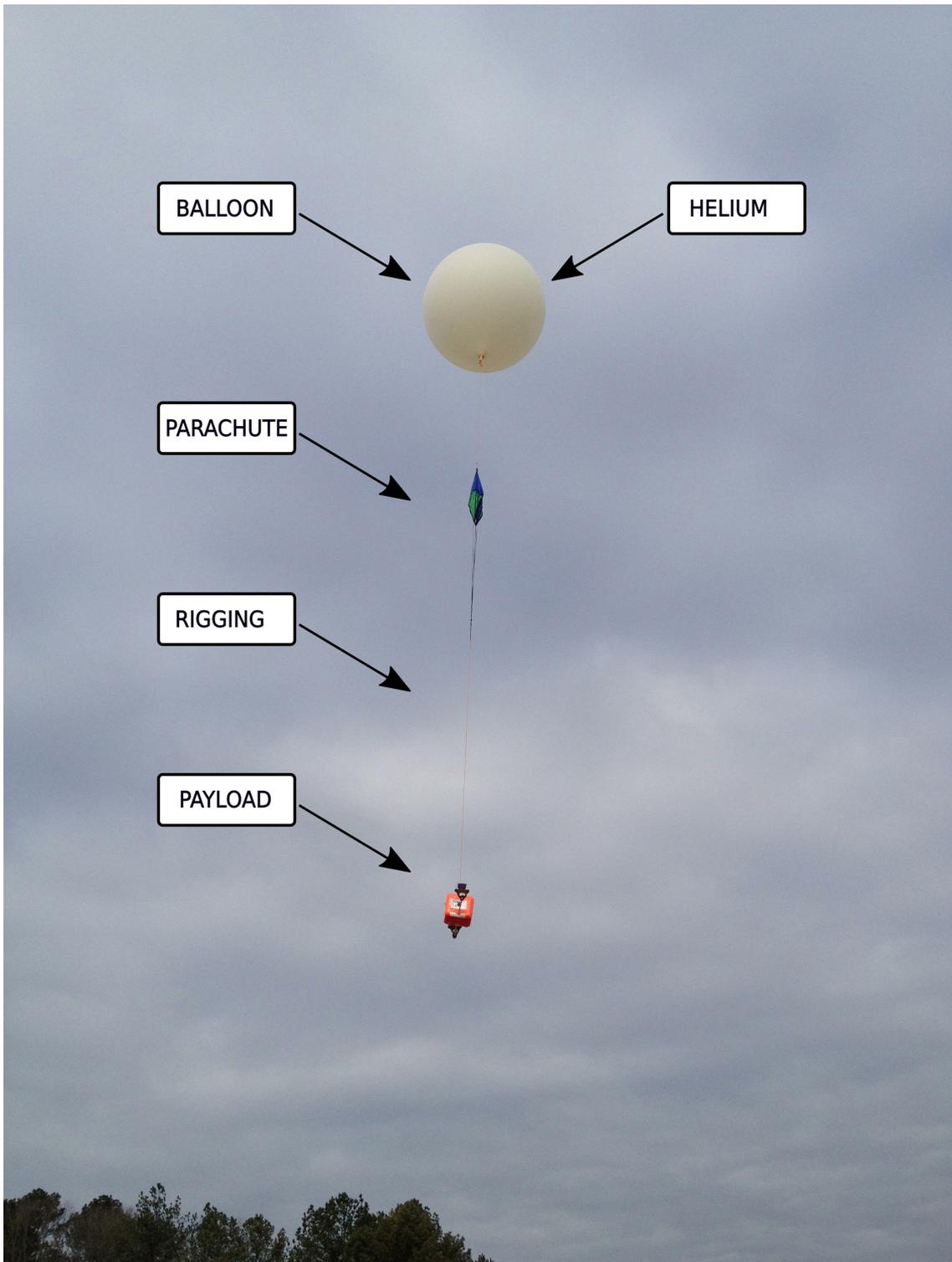
Il pallone stratosferico (figura 1.1) è un particolare tipo di pallone aerostatico, che può essere fatto di diversi tipi di materiali (si va dalla gomma naturale al lattice, dalla gomma sintetica, come il Neoprene, a tessuti in filo di Polietilene a bassa densità). Esso è ideato per il trasporto di sensori meteorologici di vario tipo e non, in ottica di osservare e studiare la Terra, per creare una rete di comunicazione o per altre applicazioni particolari, come ad esempio la sorveglianza di certe zone di particolare interesse. I palloni di questo tipo, riempiti a elio o idrogeno, raggiungono quote tra i 18 km e i 37 km circa (piena stratosfera). A quella quota, soprattutto nella parte inferiore della stratosfera, la velocità dei venti è relativamente bassa e quindi i palloni possono essere gestiti più facilmente. Sempre a quella altitudine, a causa della diminuzione della pressione esterna e quindi all'espansione dell'involucro del pallone di un fattore di circa 100:1, avviene lo scoppio e inizia la discesa. Alcuni palloni hanno raggiunto la quota da record di 53 km [29].

Si sta sviluppando una comunità scientifica mondiale, distribuita su tutti i continenti, che utilizza il pallone sonda per i propri programmi di ricerca. Questa piattaforma di palloni ad alta quota (sopra i 20 km) è quasi l'ideale per effettuare osservazioni scientifiche in un ambiente simile allo spazio, per trasportare un nuovo tipo di strumentazione innovativo e per sviluppare, in un futuro prossimo, un potenziale turismo spaziale [12]. Le discipline astrofisiche, elio-fisiche, planetarie e scienze della Terra hanno contribuito a promuovere una ripresa nell'utilizzo dei palloni stratosferici. Il crescente interesse per lo sviluppo di una solida piattaforma in grado di effettuare una grande varietà di ricerche stratosferiche ha spinto la comunità a sviluppare migliori capacità per i carichi utili in grado di volare ad alta quota per periodi più lunghi (sopra i 100 giorni), in competizione con alcune missioni spaziali [11].

### 1.2.1 Vantaggi nell'utilizzo dei palloni sonda

I vantaggi nell'utilizzare i palloni stratosferici al posto dei satelliti o dei razzi-sonda per scopi scientifici sono enumerati di seguito [2, pp. 12-14]:

1. *Basso costo*: sulla base di una prima stima del costo dell'equipaggiamento completo del pallone, esclusa la strumentazione scientifica a bordo, la spesa necessaria per un esperimento con il pallone è dieci volte inferiore rispetto a quella di un razzo-sonda, cento volte inferiore rispetto a quella di un piccolo satellite e mille volte inferiore rispetto alla spesa di un grosso satellite. Anche i costi per le strutture di lancio, per le operazioni e per la manutenzione sono molto più basse rispetto agli altri tipi di lancio.
2. *Capacità di carico alte*: i palloni possono trasportare payload scientifici che superano la tonnellata di peso alla quota di 30 km. Non hanno problemi di alloggiamento come nel caso dei satelliti, dove devono essere alloggiati nella parte anteriore del razzo. Si possono anche caricare payload molto più voluminosi.



*Figura 1.1: Pallone sonda. Vengono indicati i componenti principali del sistema. Si nota il pallone riempito ad elio, che, essendo più leggero dell'aria, spinge il pallone e il suo payload in sù. Sotto il pallone viene legato il paracadute di rientro, che dovrà aprirsi quando viene sganciato il pallone. Infine tra il paracadute e il payload vi è un insieme di cavi (rigging) che sorreggono quest'ultimo.*

3. *Recupero e riutilizzo del payload*: le attrezzature di osservazioni e di sperimentazione possono essere riutilizzate, poiché rientrano a Terra per mezzo di un paracadute e possono essere recuperati, risultando un grande vantaggio per i costi.
4. *Tempo di preparazione breve*: le vibrazioni meccaniche durante il lancio e le condizioni ambientali durante il volo per un pallone stratosferico non sono così gravi come nei casi dei razzi-sonda o satelliti. Pertanto, lo sviluppo della strumentazione può essere completato in tempi più brevi. In alcuni casi, l'equipaggiamento a bordo può essere costruito e testato direttamente dai ricercatori stessi.
5. *Alta flessibilità nella scelta del sito di lancio*: poiché le *facilities* del pallone sono relativamente piccole, essi possono essere lanciati non solo da zone che sono nelle regioni di latitudine vicine ai due tropici, come tipico per i satelliti, ma anche da zone vicino ai poli o all'equatore. Ciò significa che gli esperimenti con i palloni possono essere eseguiti in posti adatti ai requisiti fisici per l'osservazione degli obiettivi.
6. *Osservazioni dell'alta atmosfera "in situ"*: solo i palloni stratosferici sono in grado di eseguire osservazioni puntuali della stratosfera, alla quota di 30-40 km. I velivoli non possono raggiungere tali quote e i razzi sonda passano in questa zona per periodi brevi. Le osservazioni atmosferiche da un satellite orbitante è limitato a misurazioni dal punto di vista del telerilevamento (*remote-sensing*), fatte da lunga distanza.
7. *Buona funzionalità durante il volo*: eccetto casi di voli speciali a lunga distanza, i palloni, durante sperimentazioni standard, volano dove è possibile avere una comunicazione diretta con una base di Terra. Di conseguenza, sono possibili una ricezione stabile dei dati telemetrici e operazioni comandate in dettaglio, durante tutta la missione. Al contrario, nel caso di satelliti posizionati a quota media (500 km) il periodo di contatto con una singola stazione di terra è di circa 10-15 minuti, al massimo.
8. *Esperimenti per le tecnologie spaziali*: i palloni vengono utilizzati non solo per osservazioni scientifiche dello spazio, ma anche per la ricerca e sviluppo di nuove tecnologie per i razzi e i satelliti o per velivoli di rientro dall'orbita.

## 1.2.2 Regolamento

ICAO (International Civil Aviation Organization) indica le regole per l'utilizzo dei palloni stratosferici nell'Annesso 2 [10] della Convenzione sull'aviazione internazionale civile. Viene data anche la definizione del velivolo: "*Unmanned free balloon. A non-power-driven, unmanned, lighter-than-air aircraft in free flight*", cioè è un velivolo senza motore che esegue un volo libero, senza pilota ed è più leggero dell'aria. L'annesso continua esplicitando che tale velivolo deve essere utilizzato in modo da minimizzare i pericoli per persone, cose o altri velivoli. I palloni stratosferici hanno la precedenza su tutti gli altri velivoli, inclusi gli alianti. La classificazione dei palloni avviene in base al loro peso. In particolare nell'Appendice 4 dell'Annesso 2, si hanno le seguenti classi:

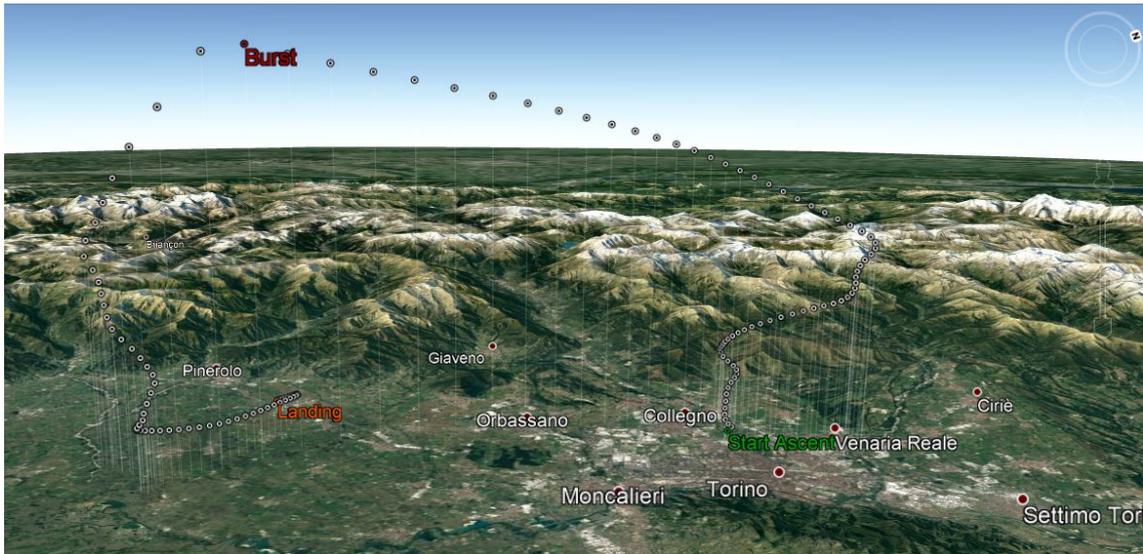


Figura 1.2: Simulazione della traiettoria di volo di un pallone stratosferico in balia del vento.

- a. *leggera*: il payload trasportato non supera i quattro kg.
- b. *media*: il payload trasportato è compreso tra i quattro e i sei kg.
- c. *pesante*: il payload trasportato è superiore a sei kg, o una combinazione di diversi tipi di imballaggio.

La classe più regolamentata è la classe pesante. Il sistema di payload in questa classe deve rispettare le norme riguardanti visibilità dei radar e le apparecchiature di comunicazione. Inoltre, le si richiede un'autorizzazione speciale per sorvolare aree soggette a restrizioni o vietate.

Il pallone sonda studiato in questa sede appartiene alla classe leggera.

### 1.2.3 Profilo di missione

La missione della piattaforma è composta di cinque macro segmenti che a loro volta possono essere suddivisi in mini-segmenti più specifici. Essi sono:

- **Preparazione e lancio**: questa fase preparatoria è composta dall'allestimento del sito di lancio, la sistemazione del payload, il controllo del corretto funzionamento dell'elettronica di bordo, eventuale simulazione della traiettoria e controllo meteorologico. Infine, si ha il lancio vero e proprio.
- **Salita**: la fase di salita avviene grazie al pallone riempito a elio. In questa fase si comincia a eseguire i vari test e osservazioni preparati per la specifica missione.
- **Galleggiamento**: questa fase, detta anche di *floating*, consiste nel mantenimento di una certa posizione (tra i 20 e i 30 km) in funzione del target da osservare o del compito da svolgere, definito anche esso in partenza. La durata di questa fase è molto variabile. In alcuni casi vi è lo scoppio del pallone e quindi l'inizio della fase successiva di discesa, in altri casi, invece, il floating perdura

e vi è uno sgancio controllato del pallone, con l'apertura di un paracadute circolare.

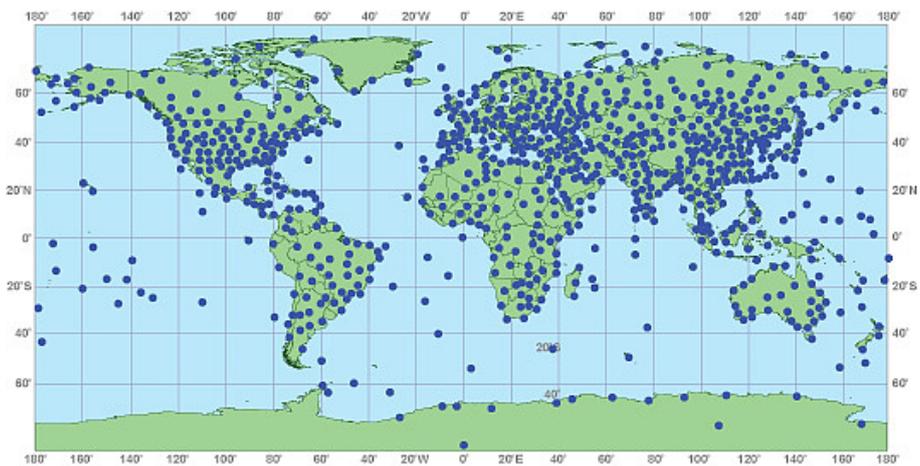
- **Discesa:** la discesa è suddivisa in due parti. Nella prima parte il sistema viene frenato dagli aerofreni e da un paracadute circolare, fino al raggiungimento di una certa quota (5-6 km), anche essa prestabilita, in cui parte la seconda fase. In questa parte, se il parafoil è stato sganciato con successo, si inizia una discesa controllata. Il controllo avviene grazie all'installazione del hardware e software a bordo della sonda. Se lo sgancio è fallito, si ha una discesa non controllata, la cui discesa è rallentata solo con i freni aerodinamici. L'utilizzo delle ventole avviene solo sotto i 3 km, poiché sopra a questa quota la pressione e la densità dell'aria sono troppo basse per permettere un corretto funzionamento delle ventole.
- **Atterraggio:** in questa macro fase si hanno l'operazione di recupero della sonda e il *post-processing* dei dati, se presenti.

Nella figura 1.2 viene indicata la traiettoria simulata del pallone. Si notano le fasi di salita, di scoppio, quindi senza floating, e la parte di discesa di una tipica missione. Il simulatore utilizzato è stato progettato sempre in *ALTEC Space*, come oggetto di tirocinio curricolare. Viene utilizzato *Google Earth Pro* [30] per la visualizzazione grafica della traiettoria. Da notare, inoltre, l'influenza dei venti sulla traiettoria. Noi ci occuperemo solo dell'ultimo segmento prima, in cui si cerca di controllare l'atterraggio.

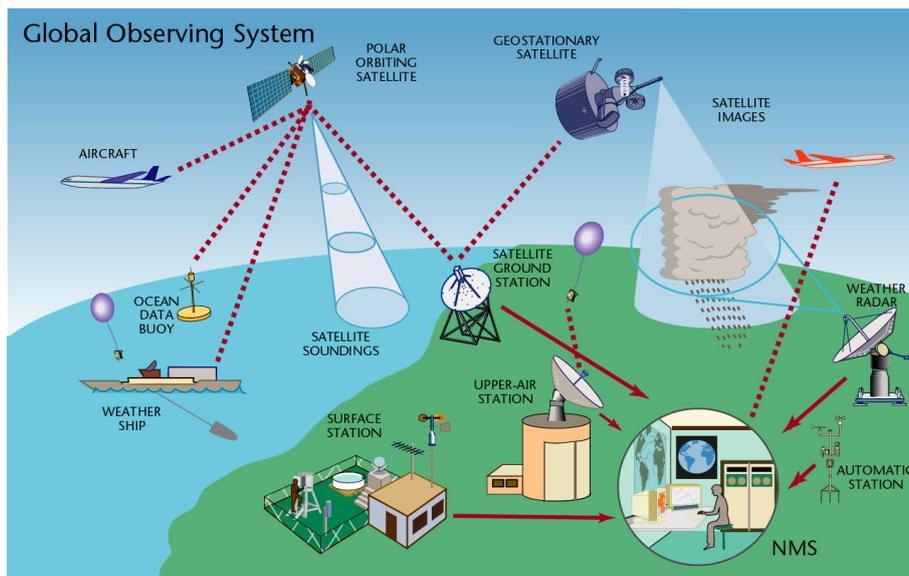
## 1.2.4 Applicazioni delle HAPS

Alcuni esempi di utilizzo delle diverse tipologie di HAP sono:

- Pallone sonda per la meteorologia: portano sensori meteorologici in quota e trasmettono a Terra i dati rilevati. Dopo un periodo di galleggiamento il pallone scoppia e la sonda rientra a Terra in caduta libera. Di questi palloni ne vengono lanciati circa 1600 al giorno da 1300 stazioni [31]. Nella figura 1.3 (a) si nota la distribuzione dei siti di lancio. Essi vengono lanciati a orari precisi (00:00UTC e 12:00UTC) e fanno parte di un sistema globale di osservazione della Terra (*GSO - Global Observing System*) [32], i cui componenti vengono indicati nella 1.3 (b).
- *HAPS (High-Altitude Platform Station)*: è una stazione su un oggetto più leggero dell'aria, ad una quota compresa tra 20 e 50 km, in un punto fisso specificato, nominale rispetto alla Terra, secondo la *ITU RR (International Telecommunication Union Radio Regulations)*. Quindi a differenza dei palloni stratosferici classici hanno un sistema propulsivo in grado di fare mantenere la propria posizione per un certo periodo di tempo. Nonostante vi siano già utilizzi pratici, è una tecnologia ancora in fase di sviluppo. Applicazioni possibili sono: telecomunicazioni, operazioni di sorveglianza e raccolta informazioni, monitoraggio in tempo reale di una regione specifica dal punto di vista sismico, meteorologico o ambientale in generale. Diversi progetti di grandi aziende sono in fase di sviluppo, come lo *Stratobus™* della *Thales Group* [33], il cui concept è indicato nella figura 1.4.



(a)



(b)

Figura 1.3: (a) Mappa mondiale delle stazioni di lancio e osservazioni delle radiosonde. (b) Componenti facenti parte del sistema globale di osservazione, gestito dalla WMO (World Meteorological Organization). Le radiosonde per l'utilizzo meteorologico sono una piccola parte dell'intero sistema.

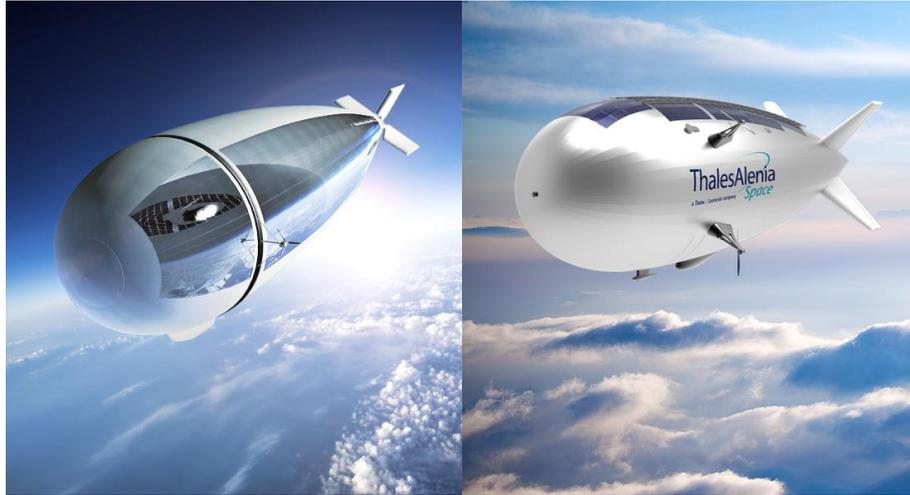


Figura 1.4: Lo Stratobus™ della Thales Alenia Space. Un concept di dirigibile come una stazione ad alta quota, un "satellite stratosferico".

- Pallone per osservazioni ed esperimenti scientifici: andando in quota si può sfruttare la riduzione dell'influenza dell'atmosfera sulla visibilità del cielo. Oltre i 30 km si hanno grandi miglioramenti nella vista di raggi cosmici, per cui si possono utilizzare la tecnologia del pallone stratosferico per analizzare fonti di raggi cosmici, come ad esempio la sorgente dei raggi X, *Cygnus X-1* [2, p. 156] od osservare e studiare lo strato dell'ozono [2, p. 160].
- Palloni per sperimentazioni ingegneristiche: alle quote in cui volano i palloni stratosferici le radiazioni provenienti dal Sole, dalla Terra, dai raggi cosmici e i gli effetti primari di questi ultimi assomigliano molto a quelli dello spazio. Per cui si possono testare tecnologie destinate all'utilizzo in orbita. È possibile simulare anche il rientro di una capsula o di un velivolo dall'orbita [2, p. 162].
- Palloni utilizzati per lanciare i razzi-sonda: sono sistemi chiamati *Rockoon*, un razzo-sonda ibrido che nella prima fase sale come se fosse un pallone stratosferico, per poi, a quote in cui la densità è molto bassa, si sgancia e vola come un razzo normale, con la possibilità di raggiungere quote più elevate. Un'azienda spagnola, la *Zero2Infinity Space* [34] vuole utilizzare questa tecnologia per mandare nell'orbita bassa piccoli satelliti.

## 2 | Progettazione

In questo capitolo vengono esposti, in ordine, il funzionamento generale del sistema di rientro, la sua configurazione iniziale, che è definita dai suoi requisiti, il hardware presente a bordo, necessario al funzionamento del sistema e la progettazione delle strutture, disegnate in CAD e costruite per mezzo della stampante 3D.

Quando è stata progettata, costruita e testata una prima configurazione, ci si è reso conto che superava poco più del 25% il peso massimo consentito dai requisiti iniziali. Il peso complessivo del hardware, incluso di cablaggio, è più di due terzi del peso complessivo che il parafoil poteva supportare. È stato necessario progettare una seconda configurazione, che fosse ancora più leggera della prima, ma che avesse le stesse funzioni. Nel capitolo le due configurazioni vengono esposte parallelamente.

### 2.1 Funzionamento del sistema

La sonda di rientro con controllo di traiettoria, dotato di un computer di bordo integrato di tipo Raspberry Pi Zero W, riceverà in ingresso dati riguardante il suo assetto e la sua posizione nello spazio grazie a dispositivi dedicati a queste due funzioni, rispettivamente: una piattaforma inerziale (IMU - Inertial Measurement Unit) e un modulo per la comunicazione con il sistema satellitare globale di navigazione (GNSS - Global Navigation Satellite System). Una volta acquisite queste informazioni il sistema dovrà essere in grado di cambiare il suo assetto e la posizione in accordo con l'obiettivo di atterraggio preposto oppure con i comandi dati dall'utente *in loco*. Il cambiamento dell'assetto, ma soprattutto della posizione avviene con la movimentazione delle linee di comando del parafoil<sup>1</sup> per mezzo dei servomeccanismi. Inoltre, vengono aggiunte due ventole inclinabili con lo scopo di migliorare la manovrabilità della sonda. I comandi dei servomeccanismi (i collegamenti fisici) passano attraverso un dispositivo chiamato Maestro, che gestisce la posizione, velocità e accelerazione con cui i servi agiscono. Il Maestro riceve a sua volta i comandi dal computer di bordo attraverso un collegamento fisico miniUSB/microUSB. La spinta dei motori viene gestita anche essa dal Maestro, il quale comunica con un controllore elettronico di velocità (Electronic Speed Control), una volta ricevuto il comando dal RPi (Raspberry Pi Zero W). Il tutto viene alimentato in maniera opportuna da un accumulatore di tipo litio-polimeri. Il peso complessivo del hardware e struttura deve aggirarsi attorno a 300 grammi, poiché è la massima portanza che può dare il parafoil a nostra disposizione.

Il linguaggio di programmazione per i software di gestione dei dati di navigazione, assetto e movimentazione dei servi è il C.

---

<sup>1</sup>Il parafoil è un paracadute in grado di creare portanza, una volta gonfiato.

## 2.2 Configurazione preliminare

Per il design preliminare si devono stabilire i requisiti fisici, soprattutto legati al peso, studiare la fattibilità dal punto di vista del hardware a bordo e, infine, disegnare il prototipo in scala della sonda.

Le due configurazioni progettate verranno trattate in parallelo in questo studio. Dal punto di vista dell'elettronica di bordo queste due configurazioni non variano. Esse si differenziano dal punto di vista strutturale e sono:

1. Configurazione con il contenitore del parafoil
2. Configurazione senza contenitore del parafoil

Quindi le variazioni sostanziali le vedremo nella parte della progettazione delle strutture.

### 2.2.1 Requisiti

I requisiti principali analizzati sono:

- **Peso:** requisito legato al parafoil utilizzato. Esso è un Opale Oxy 0.5 [35] (figura 2.1), che può reggere fino a 300 grammi. Rispetto al lavoro precedente [7], vengono aggiunte due ventole intubate di tipo *EDF-50* e due servomeccanismi per permettere la rotazione delle ventole. In questo caso il requisito di peso viene a essere molto stringente e, quindi, ad avere un impatto molto grande sulla configurazione finale della sonda.
- **Elettronica di bordo:** tutte le apparecchiature elettroniche dovranno essere alimentate in maniera corretta per la durata intera della missione.
- **Requisiti generici:** vengono tenute in conto le indicazioni dell'Annesso 2 dell'ICAO.

#### 2.2.1.1 Pesì e ingombri

I pesi e i volumi occupati dalle componenti elettroniche sono esplicitati nella tabella 2.1. Per il peso complessivo bisogna aggiungere il peso delle strutture, dei cablaggi e del materiale incollante utilizzato durante l'integrazione.

### 2.2.2 Elettronica di bordo

In questa sottosezione vengono descritti i dispositivi elettronici utilizzati, nella sottosezione successiva verrà data una stima complessiva dei costi di questi dispositivi. I componenti elettronici presenti a bordo e identici<sup>2</sup> a entrambe le configurazioni sono:

---

<sup>2</sup>Per quello che riguarda l'elettronica di bordo, l'unica differenza tra le due configurazioni è il tipo di servomeccanismo utilizzato per la movimentazione delle linee di controllo del parafoil.



*Figura 2.1: Il Parafoil utilizzato è della ditta francese Opale Paramodels, ed è un Opale Oxy 0.5. Il massimo peso che può sopportare è di 0.3 kg.*

*Tabella 2.1: Pesi e ingombri di tutti i componenti elettronici della sonda. Per le due configurazioni le componenti elettroniche rimangono le stesse. Fanno un'eccezione i servi utilizzati per il controllo del parafoil: HS-56HB nella prima configurazione e HS-85BB nella seconda configurazione. Nota importante: in questa tabella non viene preso in considerazione il peso del cablaggio.*

<b>Componente</b>	<b>Peso [g]</b>	<b>Ingombro [mm]</b>
Batteria Lipotech 1000 2S	58	11-34-69
Raspberry pi zero W v1.1	11	5-31-65
Regolatore di tensione Step-Down	9	8.8-20.3-17.8
GPS Breakout V3	8.5	25.5-35-6.5
Pololu MinIMU-9 V3	0.7	20-13-3
Micro Maestro 6-Channel	4.8	21.6-30.5-14
Regolatore per motori brushed con BEC	13	50-21-10
Ventola EDF-50	30 × 2	∅54.5-46.25
Servo Hitec HS-45HB	8 × 2	23.6-9.8-22.4
Servo Hitec HS-56HB	11.06 × 2	22.35-11.43-23.88
Servo Hitec HS-85BB	19.02 × 2	29.0-13.0-30.0

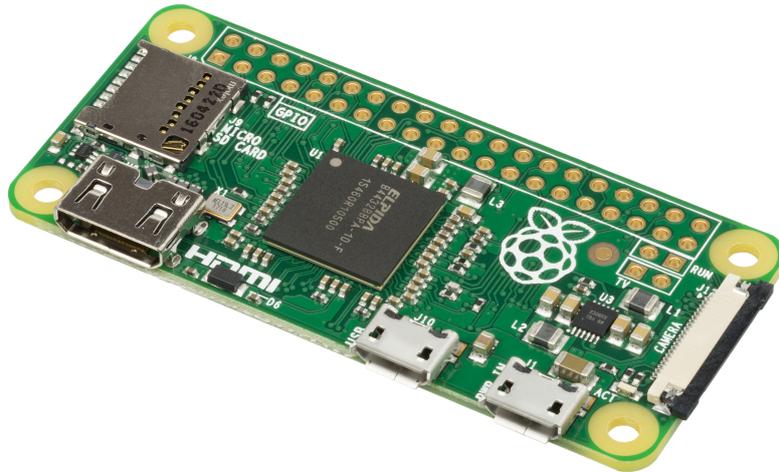


Figura 2.2: Il Raspberry Pi Zero W. È il più leggero l'on-board computer della Raspberry.

### 1 Single Board Computer - Raspberry Pi Zero W v1.1

Un computer integrato su singola scheda di dimensioni ridotte con la capacità di connettersi in rete tramite un punto di accesso *wireless*. Il meno costoso della famiglia Raspberry, esso ha le seguenti caratteristiche:

- Soc (System-on-a-chip): Broadcom BCM2835
- CPU di tipo ARM11 a 1 GHz con una RAM da 512 MB
- Wireless: 2.4 GHz 802.11n wireless LAN
- Bluetooth: Bluetooth 4.1 LE
- Alimentazione: 5V, tramite micro-USB
- Video e Audio: 1080P HD video e audio stereo tramite connettore mini-HDMI
- Memoria di massa: Scheda MircoSD da 16 GB
- Uscita: 1 micro USB, utilizzata per la comunicazione con il Maestro.
- GPIO: 40 pin, su piazzole

Il sistema operativo installatovi sopra è *Raspbian*, della famiglia Debian GNU/Linux, adattato all'utilizzo proprio sulle piattaforme Raspberry Pi. I dettagli sui software si vedranno nel capitolo quattro di questo lavoro.

A differenza dei lavori precedenti [7], in cui si è utilizzato un Raspberry Pi Model B+, di dimensione e peso più notevoli, si è preferito utilizzare questo modello più piccolo e decisamente più leggero, visibile nella figura 2.2.

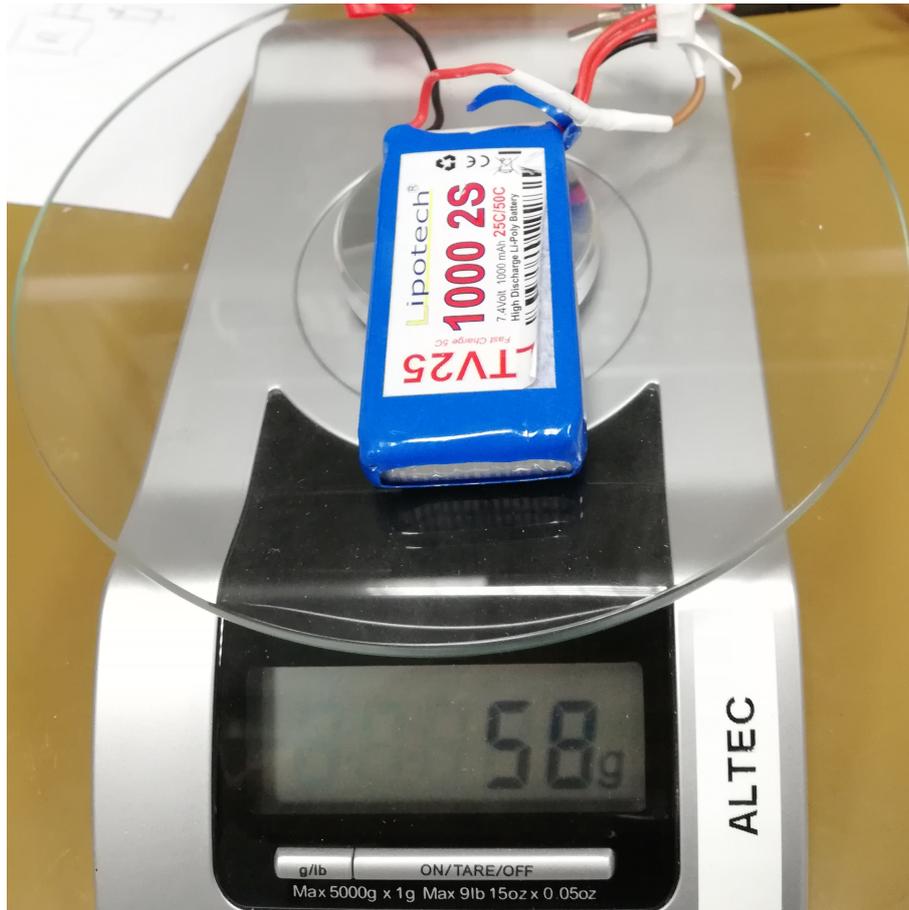


Figura 2.3: L'accumulatore litio-polimero o più comunemente LiPo utilizzato per il nostro sistema. 2 celle in serie da 3.7 V, con una capacità di 1000 mAh. Si nota il peso di 58 grammi, incluso del cablaggio.

## 1 Batteria Lipotech 1000 2S

Accumulatore litio-polimero, con una densità energetica maggiore del 20% rispetto ai più datati accumulatori agli ioni di litio, si presta perfettamente all'utilizzo dell'elettronica di bordo della nostra piattaforma. Ha una capacità di 1000 mAh, quindi a una richiesta di corrente di 1 Ampere, essa si scaricherà in 1 ora. La sigla **2S**, invece, indica due celle messe in serie. Una cella Li-Po ha un voltaggio nominale di 3.7 V, quindi la nostra batteria avrà un voltaggio nominale di 7.4 V. Nota se le celle vengono messe in parallelo (quindi vedremo la sigla P) aumenta la capacità complessiva e non il voltaggio. La scarica continua massima è di 25 A (25C nella sigla). Il peso della batteria è circa il 20% del peso totale consentito.

## 1 Regolatore di tensione Step-Down

Il dispositivo (indicato nella figura 2.4), pesante solo 3 grammi (senza i pin e i cablaggi), regola una tensione di ingresso, fino a massimo 38 V, in una tensione di uscita di 5 V e corrente di 5 A. Ha incorporato un dispositivo di protezione di inversione di tensione. Serve per alimentare il Raspberry con la tensione opportuna di 5 V.



Figura 2.4: Il regolatore di tensione di tipo step-down utilizzato nella nostra configurazione. Senza di esso non potremmo alimentare in maniera corretta il Raspberry.

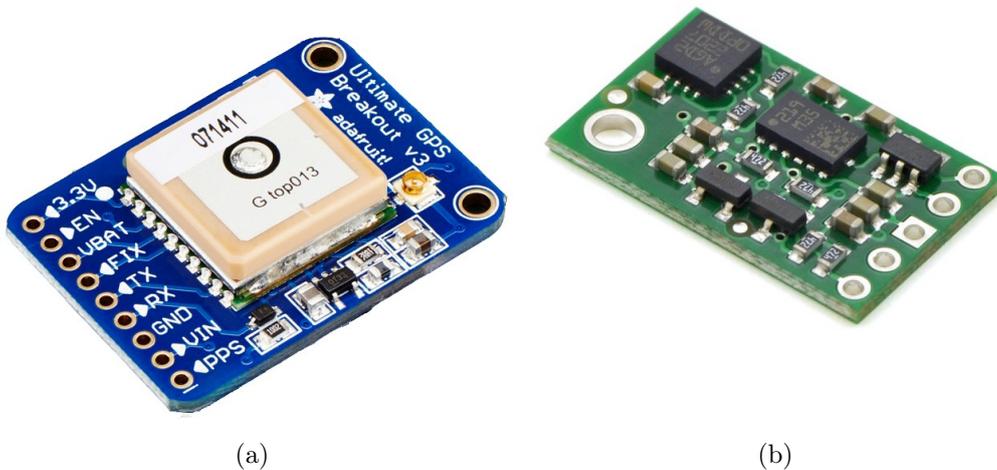


Figura 2.5: (a) Il GPS Breakout V3 della Adafruit utilizzato. (b) La piattaforma inerziale Pololu MinIMU-9 v3 utilizzata.

### 1 GPS Breakout V3

Il dispositivo della Adafruit [36], indicato nella figura 2.5 (a) è un GPS di alta qualità che può tenere traccia di massimo 22 satelliti su 66 canali. Utilizzo di corrente è molto basso (circa 20 mA durante la navigazione).

### 1 Pololu MinIMU-9 V3

La piattaforma inerziale (Inertial Measurement Unit) della Pololu [20] contiene oltre all'accelerometro e al giroscopio tri-assiali anche un magnetometro. Combinati insieme, si trova l'assetto e l'angolo di imbardata rispetto al nord magnetico. Si nota nella figura 2.5 (b).

### 1 Micro Maestro a 6 canali

Il dispositivo della Pololu [21] della figura 2.6 controlla fino a 6 servi. Quattro canali sono occupati dai quattro servi descritti sotto. Il quinto canale è occupato dal collegamento con il regolatore di velocità dei motori. Quest'ultimo, oltre a dare il segnale ai motori, alimenta il maestro stesso e di conseguenza anche i quattro servi.

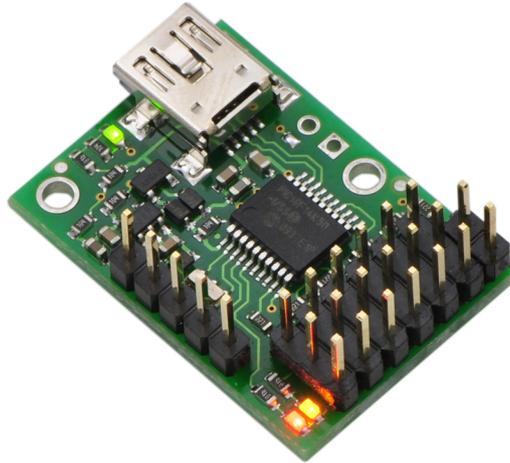


Figura 2.6: Il Maestro che controlla i servomeccanismi e il regolatore di velocità dei motori.

## 1 Regolatore per motori con BEC

L'ESC (Electronic Speed Control) è un circuito elettrico che regola la velocità di un motore elettrico (nel nostro caso i motori delle ventole intubate). Il BEC (Battery Eliminator Circuit) è un circuito che ci permette di alimentare il Maestro senza la necessità di una batteria dedicata, ma prelevando potenza dalla stessa unica batteria a disposizione. Quindi grazie a questo dispositivo (ESC con BEC) riusciamo ad alimentare sia i due motori delle ventole che il Maestro, il quale a sua volta alimenta i quattro servomeccanismi in funzione. Avere questo circuito integrato ci permette di risparmiare il peso di una batteria. Si nota tale dispositivo nella figura 2.7 (a).

## 2 Ventole EDF-50

Le ventole intubate a motore a spazzole (Electric Ducted Fan) (figura 2.7 (b)) con voltaggio di 7.2 V generano una spinta di 54 grammi ciascuna con un utilizzo di 2.62 A. Il peso totale di entrambi i fan è di circa 60 grammi, tolto il cablaggio.

## I Servomeccanismi

Abbiamo a disposizione tre tipi di servomeccanismi, indicati nella figura 2.8. Essi sono:

- 2 Servomeccanismi HS-45HB: questi servi, tra i più piccoli della loro categoria, creano una coppia di circa 1 kg·cm, pesando solo 8 grammi, senza cablaggio. Sono stati utilizzati per la movimentazione delle ventole intubate, per entrambe le configurazioni studiate. Tensione di esercizio: 4.8 V - 6.0 V; Range di massima Coppia: 1.0 - 1.2 kg/cm; Assorbimento di corrente in stallo: 440 mA.
- 2 Servomeccanismi HS-56HB: questi servi sono stati utilizzati nella prima configurazione per la movimentazione dei fasci di controllo del parafoil. Tensione di esercizio: 4.8 V - 6.0 V; Range di massima Coppia: 1.2 - 1.4 kg/cm; Assorbimento di corrente in stallo: 700 mA.
- 2 Servomeccanismo HS-85BB+: questi servi sono stati utilizzati nella seconda configurazione per la movimentazione dei fasci di controllo del parafoil. Sono

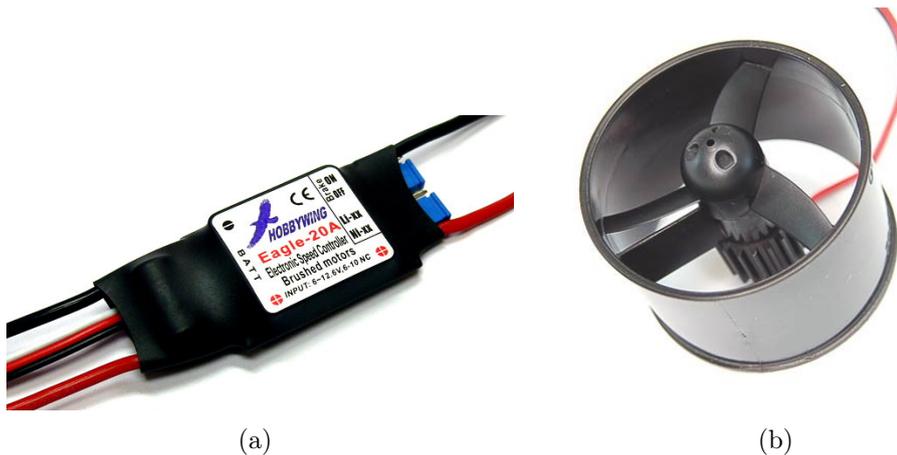


Figura 2.7: (a) Il regolatore elettronico di velocità, compreso di BEC (Battery Eliminator Circuit), che permette di alimentare i servomeccanismi senza la necessità di una batteria dedicata, ma usando la stessa che alimenta i motori delle ventole intubate. Questo tipo di regolatore, in particolare, è progettato per i motori a spazzola, si notino i due cavi che vanno ai motori e non tre, come nel caso di motori senza spazzole. (b) Ventola intubata utilizzata. Sono state utilizzate due di queste ventole con motori a spazzole.

più grandi e più potenti di quelli utilizzati nella prima configurazione 4.8 V - 6.0 V; Range di massima Coppia: 3.0 - 3.5 kg/cm; Assorbimento di corrente in stallo: 1200 mA.

### 2.2.3 Costo dell'elettronica di bordo

I dispositivi sono stati acquistati on-line da negozi autorizzati a prezzo di listino. Nella tabella 2.2 vengono indicati i costi medi dell'hardware impiegato nella seconda configurazione. Abbiamo considerato la seconda configurazione perché costa un po' di più a causa dei servomeccanismi più grandi utilizzati. La tabella non indica il costo delle strutture, cavi, pin e materiale ausiliario utilizzato per costruire tutto il sistema. Inoltre non vengono considerati i costi di spedizione, che possono impattare di molto sul costo finale.

Al costo dell'elettronica di bordo, va aggiunto il costo del parafoil, che si aggira intorno a 200 € e il costo della struttura, che viene calcolato in base al materiale utilizzato e alla modalità di stampa. Questo costo, insieme al costo del cablaggio, non dovrebbe superare 10 €.

## 2.3 Componenti Strutturali

I componenti strutturali progettati e di seguito stampati sono il risultato di un compromesso tra robustezza e leggerezza. Essi verranno in seguito costruiti utilizzando una stampante 3D.

Per la prima configurazione abbiamo:

- **Contenitore del parafoil:** un cilindro aperto a un'estremità, dentro il quale alloggia il parafoil durante la fase di salita e di floating. Al fondo del cilindro



(a)



(b)



(c)

*Figura 2.8: (a) Servo HS 45-HB per la movimentazione delle ventole, utilizzato per entrambe le configurazioni. (b) Servo HS 56-HB per la movimentazione del parafoil, per la prima configurazione. (c) Servo HS 85-BB+ per la movimentazione del parafoil per la seconda configurazione.*

Tabella 2.2: Costo medio del hardware per la seconda configurazione. Per il costo totale bisognerà aggiungere il costo delle strutture, dei cavi, dei pin, della colla e del nastro adesivo, che hanno, tuttavia, un impatto minore rispetto all'elettronica di bordo qui presentata. Il costo del Parafoil, non indicato nella tabella, è di 200 euro.

Componente	N°	[€]	Nota
Batteria Lipotech 1000 2S	1	6	
Raspberry pi zero W v1.1	1	26	Starter Kit compreso
Scheda di Memoria MicroSDHC	1	11	16 GB
Regolatore di tensione Step-Down	1	15	
GPS Breakout V3	1	35	
Pololu MinIMU-9 V3	1	13	Non più disponibile
Micro Maestro 6-Channel	1	17	
Cavo microUSB/miniUSB	1	4	Collegamento con RPi
Regolatore per motori brushed con BEC	1	15	
Ventola EDF-50	2	40	L'unità costa circa 20.00 €
Servo Hitec HS-45HB	2	30	L'unità costa circa 15.00 €
Servo Hitec HS-85BB+	2	36	L'unità costa circa 18.00 €
<b>Totale</b>		<b>248</b>	Senza costi aggiuntivi

vi è attaccata una molla che permette di espellere il parafoil al momento opportuno e quindi iniziare la discesa controllata. Il sistema di espulsione non è stato approfondito in questo lavoro.

- **Anello superiore:** una struttura circolare che va ad alloggiarsi intorno al cilindro nella parte superiore. Esso dovrà supportare i quattro servomeccanismi e i le due ventole intubate.
- **Contenitore Raspberry:** questa struttura, rettangolare esternamente, viene incollata su un lato del cilindro, di cui ne segue il profilo. Conterrà il Raspberry e il Maestro.
- **Contenitore Batteria e ESC:** simile a quello del Raspberry, ma più grande, viene incollato sul lato diametralmente opposto. Dovrà contenere la batteria, il controllore di velocità e il convertitore di tensione.
- **Struttura di irrobustimento:** questa struttura in polistirolo viene posta intorno al contenitore centrale per irrobustire e proteggere in caso di atterraggio l'elettronica di bordo

Per la seconda configurazione, in cui si è cercato di ridurre ulteriormente il peso, si ha:

- **Supporto Superiore:** è una struttura simile all'anello superiore della prima configurazione, ma più piccola e più piena, dal momento che non deve circondare il contenitore del parafoil. Su questa struttura vengono fissati il GPS, l'IMU e il Maestro nella parte centrale

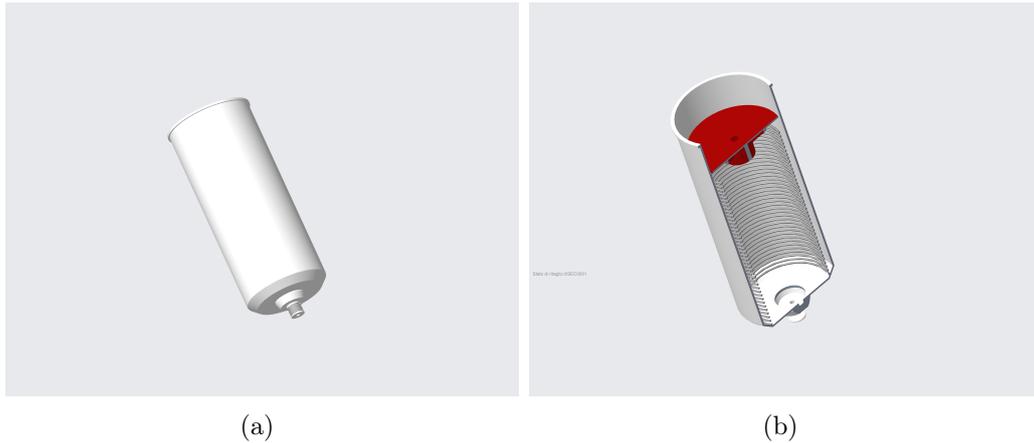


Figura 2.9: (a) il contenitore che contiene il parafoil nella fase di salita e di osservazione. (b) il contenitore sezionato, con molla rilasciata.

- **2 Supporti Lateral**i: queste 2 strutture speculari, hanno il compito di agganciarsi al supporto superiore e di reggere ognuno i due servi e la ventola intubata.
- **Struttura Inferiore**: un blocco di polistirolo incavato da entrambi i lati da permettere l'alloggiamento della batteria, dell'ESC e del Raspberry Pi Zero W.

## 2.4 Disegno CAD

Per il disegno è stato utilizzato il software di modellazione 3D Creo Parametric 4.0 [5].

### 2.4.1 Configurazione 1

#### 2.4.1.1 Contenitore Parafoil

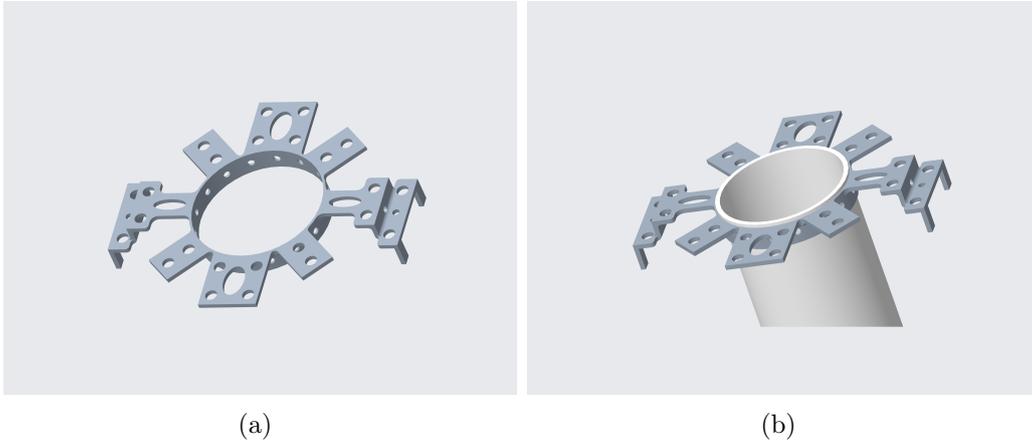
Il contenitore del parafoil (figura 2.9) ha la funzione di contenere il paracadute nella fase di salita e di floating. Sin da subito si aveva a disposizione tale contenitore e si è pensato di costruire tutta la struttura intorno a questo componente.

#### 2.4.1.2 Anello superiore

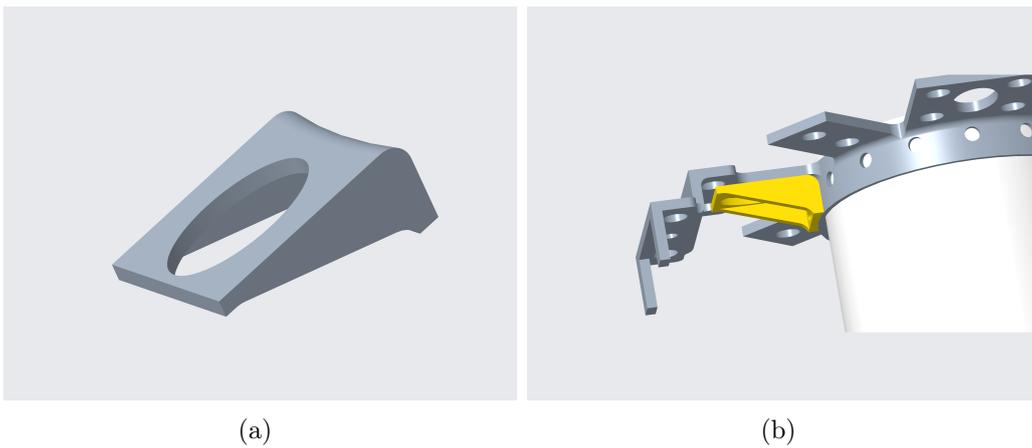
Questo componente (figura 2.10) è quello che regge i quattro servomeccanismi e le due ventole. Esso è il più sollecitato e, infatti, dopo aver fatto dei test di sollecitazione di flessione, è stato necessario rafforzare la struttura con delle controventure (figura 2.11), stampate in un secondo momento. Di seguito si sono fatti ulteriori test e lo spostamento trasversale della struttura si era quasi annullato.

#### 2.4.1.3 Contenitore Raspberry

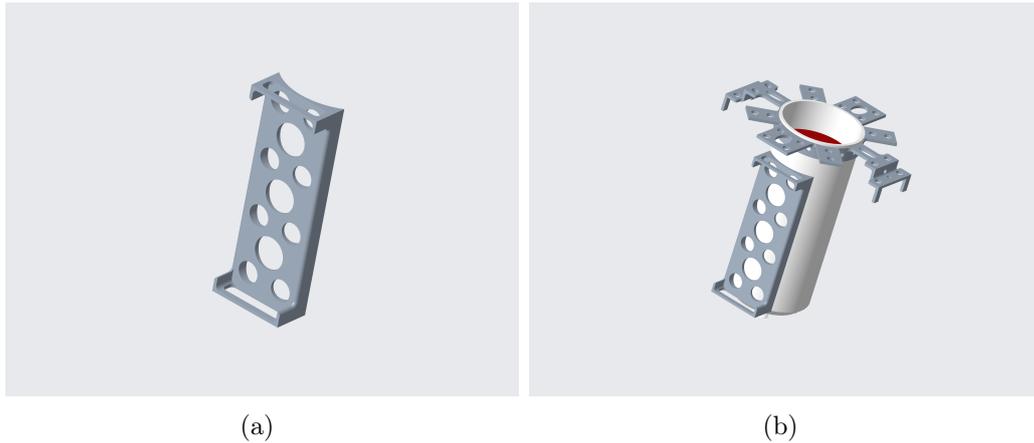
Questo supporto (figura 2.12) è uno dei più leggeri. Esso ha dalla parte esterna la superficie dritta che permette l'appoggio del Raspberry e del Maestro, invece, dalla parte interna, quella rivolta al contenitore del parafoil, esso si curva, seguendo la



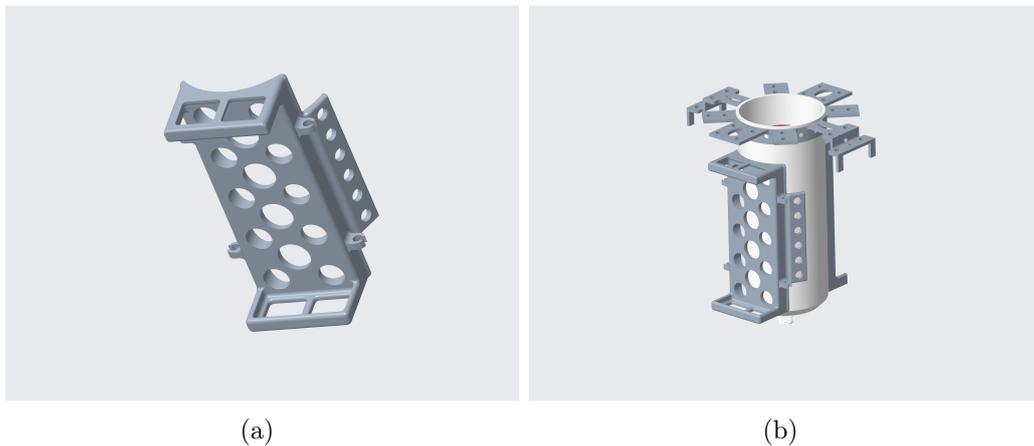
*Figura 2.10: (a) l'anello superiore. (b) l'anello superiore assemblato sul contenitore.*



*Figura 2.11: (a) Controventatura. (b) Posizionamento di tale supporto sulla parte più sollecitata, in seguito a dei test di sollecitazione a flessione.*



*Figura 2.12: (a) Contenitore del On Board Computer Raspberry Pi Zero W v1.1 e del Maestro Pololu. (b) Posizionamento di tale supporto sulla parete del contenitore centrale. Si è cercato di lasciare un margine molto piccolo nelle strutture rispetto alle dimensioni dell'elettronica di bordo.*



*Figura 2.13: (a) Contenitore della batteria e del controllore di velocità dei motori. (b) Posizionamento di tale supporto sulla parete del contenitore centrale.*

curvatura del tubo centrale. Per evitare di costruire un apposito contenitore per il Maestro, si è deciso di alloggiare anche quest'ultimo dentro questo unico supporto, allungandolo solo dei millimetri necessari.

#### **2.4.1.4 Contenitore Batteria**

Il contenitore della batteria (figura 2.13) è il più pesante delle componenti strutturali progettate. Esso deve contenere la batteria e l'ESC, insieme al regolatore di tensione DC-DC di tipo step-down. Simile al contenitore della scheda del Raspberry e del Maestro, la struttura è più evoluta, con due alette laterali da permettere l'ancoraggio dell'ESC, ma come si vedrà in seguito si è deciso di incollarlo direttamente sulla batteria. È stato lasciato spazio anche per il regolatore di tensione, necessario, poiché il Raspberry va alimentato ad una tensione più bassa rispetto a quella erogata dalla batteria.

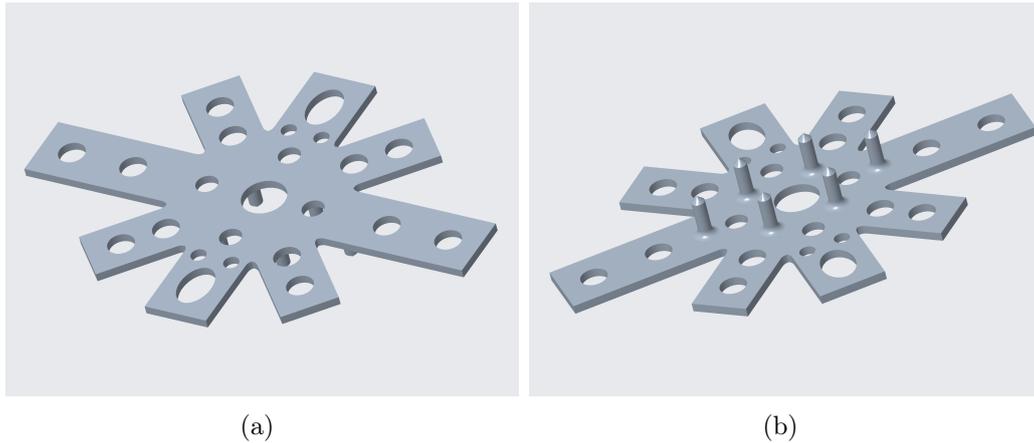


Figura 2.14: Supporto superiore: (a) vista dall'alto, (b) vista dal basso. Si notano i denti nella parte inferiore che andranno a conficcarsi dentro la struttura del basamento. Verrà utilizzata della colla per rendere più saldo il collegamento.

## 2.4.2 Configurazione 2

La seconda configurazione, come accennato, è più leggera e di più semplice costruzione, in quanto manca del tutto l'alloggiamento del parafoil, che verrà quindi portato esteriormente.

### 2.4.2.1 Supporto Superiore

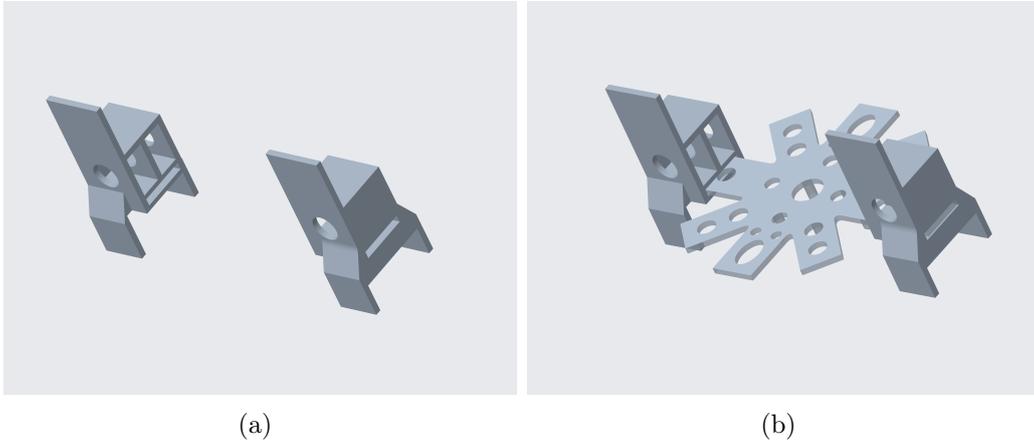
Questo componente (figura 2.14) riprende parzialmente la forma dell'anello superiore della prima configurazione, ma non più con un foro centrale che permetteva l'alloggiamento intorno al contenitore del parafoil, ora non più necessario. Si risparmiano volumi e pesi. Nella posizione centrale, ora liberatasi, si è pensato di mettere in seguito il Maestro, avvicinandolo ai servi e quindi la possibilità di accorciare i cavi che li collegano. Su questo supporto verranno incollati anche il ricevitore GPS e la piattaforma inerziale. Ad esso vanno collegati i supporti dei servi.

### 2.4.2.2 Supporti dei servi

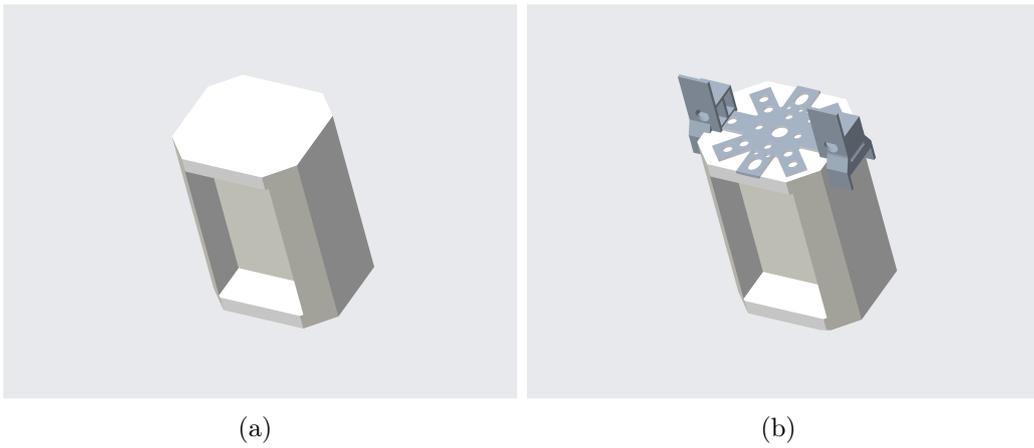
Queste due componenti strutturali (figura 2.15) sono speculari una rispetto all'altra. Ognuna di esse ha la funzione di alloggiare il servo del parafoil, il servo che fa muovere la ventola e la ventola stessa. Viene creato un'asola per ognuno di loro, dentro il quale viene inserito un braccio del supporto superiore.

### 2.4.2.3 Supporto inferiore

Questo supporto (figura 2.16) è fatto in polistirolo per abbassare di più il peso complessivo delle strutture, e ha la funzione di alloggiare la batteria, il Raspberry, il controllore di velocità dei motori e il regolatore di tensione. Sono stati analizzati due tipi di questo basamento: uno fatto a doppia T e uno a forma di prisma, con incavati degli alloggiamenti per l'elettronica che deve contenere. È stata scelta, alla fine, questa seconda configurazione perché più robusta, a scapito di un piccolo incremento di peso. È importante sottolineare come la base della seconda configurazione, fatta in polistirene (densità del PS è intorno a  $1.050 \text{ g/cm}^3$ ) sia più leggera rispetto alla



*Figura 2.15: (a) supporti ausiliari per i servi; (b) vista assemblata con il supporto superiore.*



*Figura 2.16: (a) Supporto inferiore; (b) Assieme di tutti i componenti strutturali.*

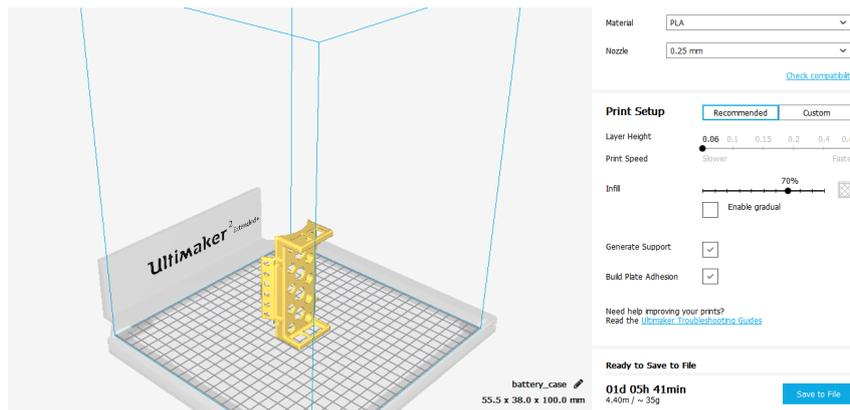


Figura 2.17: Esempio di schermata del software *Ultimaker Cura*. Da notare come viene indicato anche il tempo necessario a eseguire la stampa, la quantità di materiale utilizzato e quindi il peso finale in grammi.

base della prima configurazione, fatta in polivinilcloruro (densità del PVC è di circa  $1.40\text{-}1.45\text{ g/cm}^3$ ), a cui vanno aggiunti il peso della molla di estrazione del parafoil e la base della molla. I volumi occupati per le due configurazioni sono circa gli stessi.

## 2.5 Stampa 3D

Il software utilizzato per preparare i file CAD alla stampa è *Ultimaker Cura 3.4.0* [6]. L'utilizzo di questo software è concesso a titolo gratuito dalla casa costruttrice delle stampanti 3D [39]. Una tipica schermata è indicata nella figura 2.17. Si è utilizzato il macchinario *Ultimaker 2 Extended+* [1]. Dal software si sceglie la macchina che si vuole utilizzare e i parametri vengono impostati automaticamente.

Il materiale utilizzato per la stampa delle strutture è il PLA (acido polilattico). La densità è di circa  $1.24\text{ g/cm}^3$ , meno pesante del PVC, ma più pesante del PS. La temperatura ottimale di stampa è intorno a  $215\text{ }^\circ\text{C}$ . Il materiale si presenta sotto forma di un rotolo di filamento lungo qualche decina di metri. La dimensione della sezione del filamento è di circa  $2.85 \pm 0.10\text{ mm}$

La dimensione del diametro dell'ugello utilizzato è di  $0.25\text{ mm}$ .

La preparazione dei file avviene nei seguenti passaggi:

1. Dal software CAD (qui *Creo Parametric*) si esporta il disegno dell'elemento nel formato *.stl* (*Standard Triangulation Language*<sup>3</sup>)
2. Il file nel formato *.stl* viene importato nel software *Ultimaker Cura* e si impostano i parametri di stampa, quali ad esempio:
  - il tipo di materiale (PLA)
  - il diametro dell'ugello ( $0.25\text{ mm}$ )
  - l'altezza dello strato o la velocità di stampa
  - percentuale di riempimento ( $70\%$ )
  - generare o meno i supporti ausiliari

<sup>3</sup>.stl è un formato di file binario o ASCII che rappresenta un solido la cui superficie è stata discretizzata in triangoli

Tabella 2.3: *Pesi e ingombri delle componenti strutturali della sonda. Nota: in questa tabella non viene preso in considerazione il peso delle strutture di supporto in polistirolo.*

	<b>Componente</b>	<b>Peso [g]</b>	<b>Ingombro [mm]</b>
Configurazione 1			
	Contenitore Parafoil	36	56-56-143
	Anello superiore	13	20-110.6-114.6
	Contenitore batteria	18	100-37.8-54.95
	Contenitore Raspberry	11	100-35.38-17.8
	Leva parafoil	1 × 2	32.5-1.5-12.5
	Collegamento ventola	1 × 2	30-32-9
Configurazione 2			
	Supporto superiore	8	107-75-12
	Supporto servi	8 × 2	29-56.5-20
	Leva parafoil	1 × 2	32.5-1.5-12.5
	Collegamento ventola	2 × 2	34-39-13

- generare o meno uno strato sottile di adesione alla base
  - altri parametri personalizzabili
3. Si esporta il file così modificato in un nuovo formato `.gcode`<sup>4</sup>
  4. Il file `.gcode` viene letto dalla stampante 3D attraverso una scheda di memoria di tipo SD.

Nella tabella 2.3 vengono indicati gli ingombri e i pesi delle componenti strutturali. Nella figura si osserva un esempio di come si presentano i pezzi appena stampati, in particolare sono alcuni pezzi della seconda configurazione: i due bracci per la movimentazione del parafoil, i supporti per i servi e i collegamenti tra le ventole e i servi.

---

<sup>4</sup>Il file nel formato `.gcode` è un file contenente una sequenza di comandi che descrivono il lavoro da compiere da parte della stampante 3D.

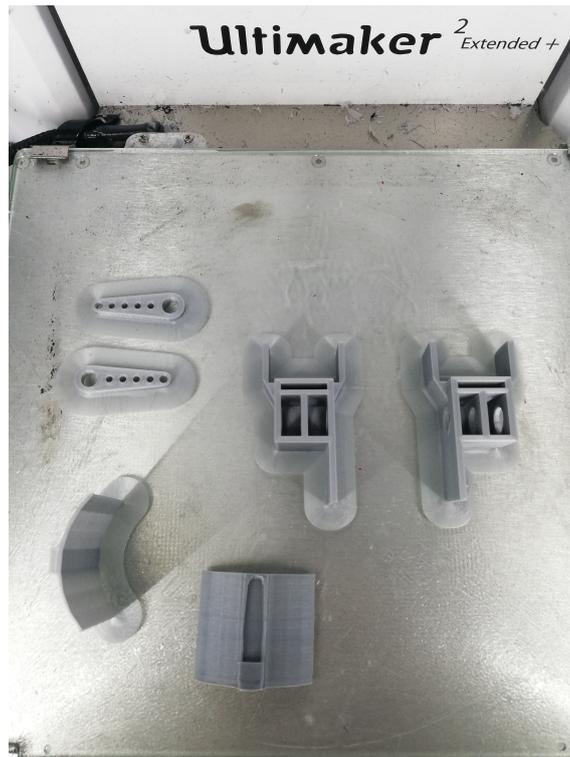
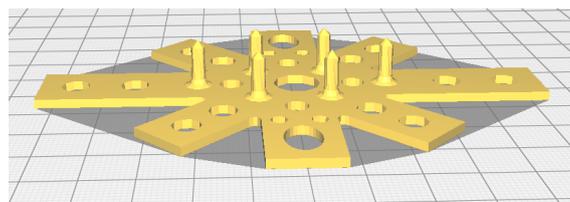


Figura 2.18: Alcuni pezzi della seconda configurazione appena stampati. Si nota come viene stampata anche uno strato (brim in inglese) che permette una miglior adesione al piattello inferiore del pezzo in lavorazione. A stampa completata, lo strato sottile può essere separato dal modello solido e quindi scartato.



(a)



(b)

Figura 2.19: Supporto superiore della seconda configurazione: (a) Vista in Ultimaker Cura; (b) Supporto stampato. Si nota sul bordo inferiore piccole irregolarità dovute allo strato di adesione iniziale (brim) che permette una stampa più stabile.

## 3 | Integrazione

Nella prima parte del capitolo si discute sul collegamento elettrico tra i dispositivi, nella seconda, invece, su come è stata effettuata l'integrazione dell'elettronica con la struttura.

### 3.1 Integrazione dell'elettronica di bordo

In questa sezione si discute sull'integrazione dell'elettronica di bordo, in cui vengono inclusi lo schema elettrico del hardware, il tipo di collegamento fisico tra i vari dispositivi, la durata della batteria.

#### 3.1.1 Schema elettrico

Nella figura 3.1 viene indicato lo schema elettrico generale della sonda. La batteria, attraverso un interruttore alimenta, da una parte il controllore elettronico di velocità (ESC) che può ricevere in ingresso il voltaggio della batteria, dall'altra il Raspberry, mettendogli a monte un convertitore di tensione in corrente continua poiché esso può ricevere in ingresso massimo 5 V. Il Raspberry alimenta a sua volta il GPS e la piattaforma inerziale che gli forniscono i dati di posizione e di assetto e comunica con il Maestro attraverso un cavo mini-USB/micro-USB. L'alimentazione del Maestro non avviene per mezzo di questo cavo USB, bensì attraverso il cavo dei dati dell'ESC che grazie alla presenza del circuito BEC, permette di alimentare i servomeccanismi connessi al Maestro senza dover dedicarli una batteria apposita.

Nella tabella 3.1 vengono indicati i consumi in termini di corrente assorbita e i voltaggi nominali di funzionamento. Da questi dati, combinati insieme ai dati sulla capacità della batteria, si possono calcolare la durata della vita media di tutta l'elettronica di bordo. In questo modo possiamo vedere in prima ottica la durata della missione per questo prototipo in condizioni nominali oppure estreme.

#### 3.1.2 Collegamento tra i dispositivi

Il collegamento dei nostri dispositivi avviene nella maggior parte dei casi grazie all'utilizzo dei cavi elettrici standard che vanno a collegarsi ai pin saldati a stagno sui seguenti dispositivi: Raspberry, Maestro, GPS, IMU.

I **cavi sottili**, indicati nella figura 3.2, utilizzati per il passaggio dei dati e di piccole correnti hanno le seguenti caratteristiche:

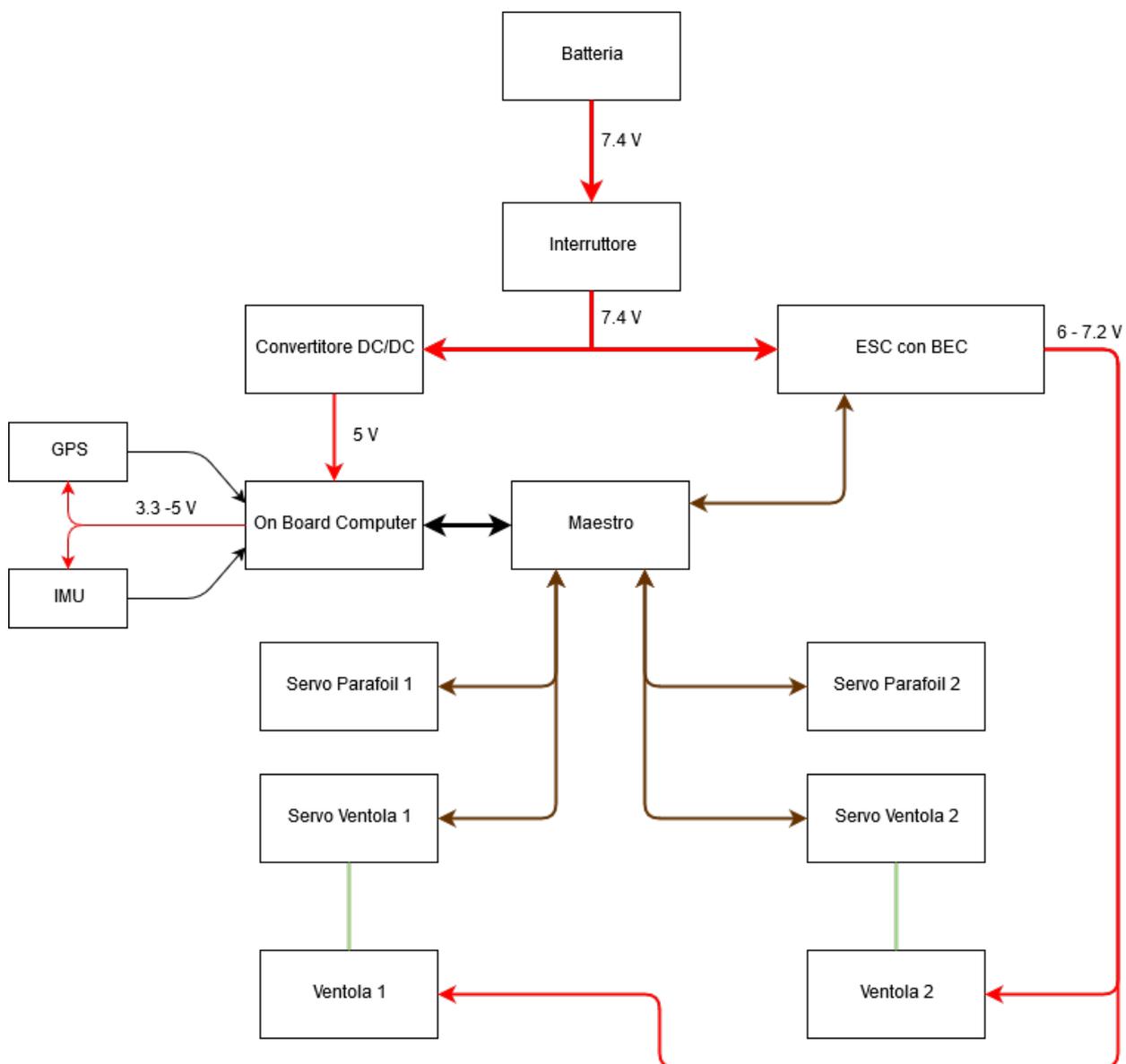


Figura 3.1: Schema elettrico generale del sistema. Le frecce rosse indicano l'alimentazione, le nere il passaggio di dati, le marroni sia alimentazione che dati, le verdi collegamento meccanico.

Tabella 3.1: Corrente assorbita e voltaggi nominali di funzionamento dell'apparecchiatura elettronica a bordo. Si ricorda che la batteria utilizzata è una a litio-polimeri da 1000 mAh di capacità.

Comp.	Corr. Ass. [mA]	Volt. [V]	Nota
Raspberry Pi Zero W	180	5.0	Alimenta GPS e IMU
Regolatore di tensione	1	6.0-38	Regola la tensione a 5 V
GPS	25	3.0-5.5	
IMU	6	2.5-5.5	
Maestro	50	5.0-16	
ESC con BEC		7.4	Eroga fino a 20 [A]
Ventola EDF50	2620-3390	6.0-7.2	Spinta generata 54-72 g.
Servo Hitec HS-45HB	440	4.8-6.0	
Servo Hitec HS-56HB	550	4.8-6.0	Prima config.
Servo Hitec HS-85BB	1200	4.8-6.0	Seconda config.

- AWG<sup>1</sup>: 26
- Range di temperatura: -30/+80 °C
- Conduttore: rame stagnato
- Diametro: 0.405 mm
- Voltaggio: 300 V
- Isolante: PVC
- Corrente di fusione: 20 Ampere per circa 10 secondi; 37 Ampere per 1 secondo; 208 Ampere per 32 ms

Tenendo in mente la figura 3.1 consideriamo i collegamenti per ognuna delle apparecchiature elettroniche che hanno il ruolo di gestione dei dati: il Raspberry e il Maestro. In particolare, al primo vengono attaccati il GPS e la piattaforma inerziale, mentre al secondo vengono attaccati tutti i servomeccanismi e il cavo di alimentazione e passaggio dei dati proveniente dall'ESC.

### 3.1.2.1 GPS e IMU collegati al Raspberry Pi Zero

Raspberry Pi Zero è in grado di comunicare con altri dispositivi attraverso pin specifici e nel nostro caso, prima di collegare i cavi fisicamente, è bene capire il posizionamento e come funzionano tali collegamenti. Dopo averli collegati, vi è la necessità di configurare anche le librerie del Raspberry Pi per poter trasmettere i dati effettivamente: di questo aspetto ci occuperemo nelle prossime sottosezioni. In particolare:

<sup>1</sup>AWG sta per **American Wire Gauge** ed è un sistema standard per misurare la sezione dei fili, che determina la massima intensità di corrente che può gestire. Dal numero indicato si possono ricavare tutti le proprietà geometriche e di resistenza del filo.

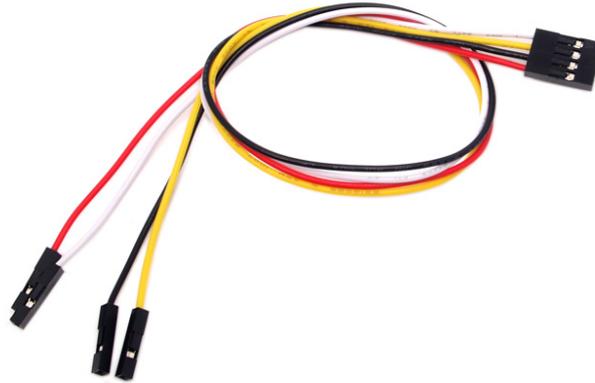


Figura 3.2: La tipologia di cavo utilizzato. Per il collegamento dei singoli dispositivi, a cui sono stati saldati i pin, si utilizzano cavi con uscite Femmina-Femmina. In pochi casi sono stati utilizzati cavi con uscite Maschio-Femmina o Maschio-Maschio.

- GPS: il Raspberry Pi Zero W ha la possibilità di comunicare con dispositivi che supportano l'utilizzo dell'**UART** (Universal Asynchronous Receiver-Transmitter), presente sul nostro GPS Breakout V3. È un circuito fisico integrato con un micro-ctrllore e ha il compito di mandare e ricevere dati seriali. Il terminale del trasmettitore del primo dispositivo va a collegarsi al terminale di ricevitore del secondo e viceversa:  $T_{X_1} \rightarrow R_{X_2}$  e  $T_{X_2} \rightarrow R_{X_1}$ , dove 1 è il primo dispositivo e 2 è il secondo dispositivo. Vanno aggiunti i cavi di alimentazione e di messa a terra.
- IMU: in questo caso il Raspberry Pi può utilizzare un protocollo seriale chiamato I<sup>2</sup>C (Inter Integrated Circuit) di cui la piattaforma inerziale della *Pololu* è in possesso. I<sup>2</sup>C, essendo un protocollo seriale, utilizza solo due linee. Esse sono SDA (Serial Data) per i dati e SCL (Serial Clock) per il clock. Da questo ultimo collegamento si nota che I<sup>2</sup>C è un bus sincrono.

Nella figura 3.3 vengono indicati schematicamente i collegamenti descritti sopra (tra il Raspberry, il GPS e la piattaforma inerziale).

### 3.1.2.2 Servomeccanismi ed ESC collegati al Maestro

I servomeccanismi e l'ESC si presentano con i cavi già incorporati e quindi è sufficiente collegarli direttamente al Maestro. I cavi in questione sono tre e sono collegati insieme: normalmente si ha un cavo di colore rosso per l'alimentazione, un cavo nero per la messa a terra e un cavo di colore chiaro (bianco o giallo) per i dati. Attraverso questi cavi quindi si alimentano e si comandano i servomeccanismi. Il cavo proveniente dall'ESC alimenta in parallelo tutti i servomeccanismi, grazie al circuito integrato BEC (Battery Eliminator Circuit). Nella figura 3.4 viene schematizzato il collegamento fisico tra i vari dispositivi in questione. Il Maestro, per ricevere i comandi esterni, si dovrà collegare al Raspberry Pi Zero grazie a un cavo micro/mini USB, non presentato nello schema.

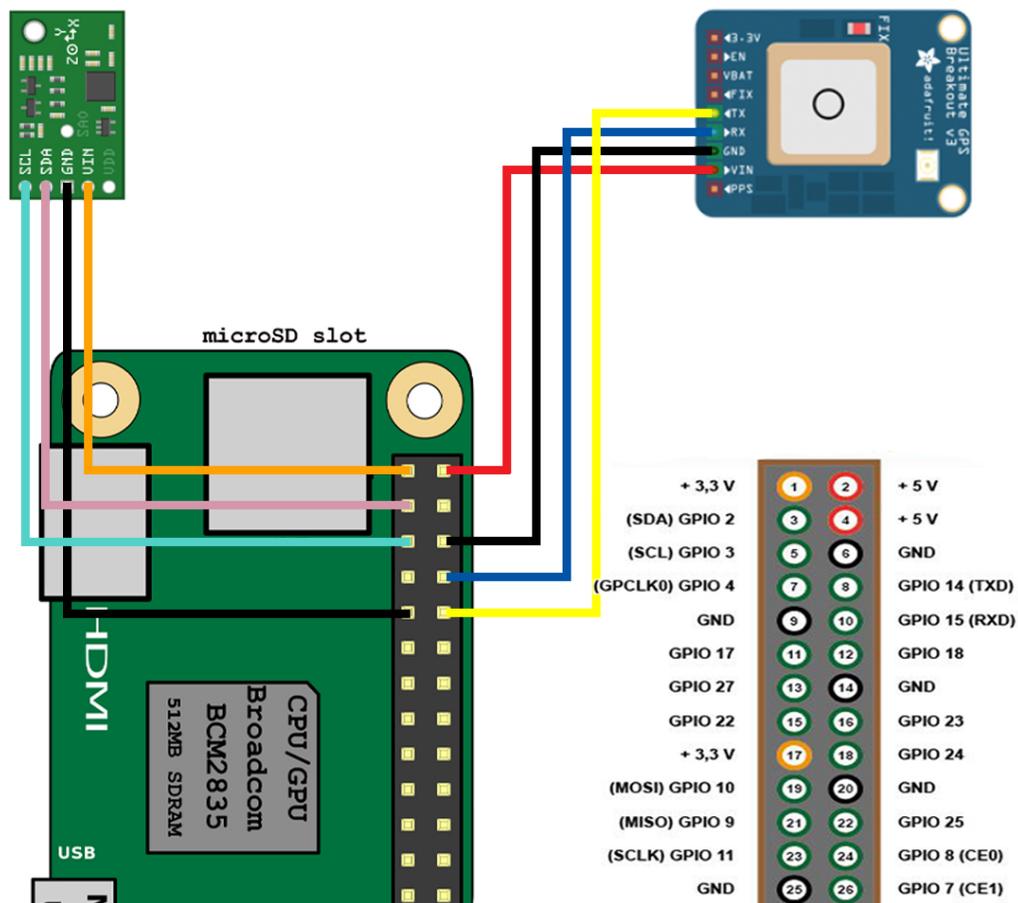


Figura 3.3: Il collegamento tra il Raspberry Pi, GPS e IMU. Viene mostrata solo la parte superiore del RPi. Di lato vengono indicati come possono essere utilizzati i pin e quindi in che maniera vanno collegati i fili provenienti dai sensori. In tutto avremo quattro cavi per ogni dispositivo.: Tx, Rx, Vin, GND per il GPS e SDA, SCL, Vin, GND per la piattaforma inerziale.

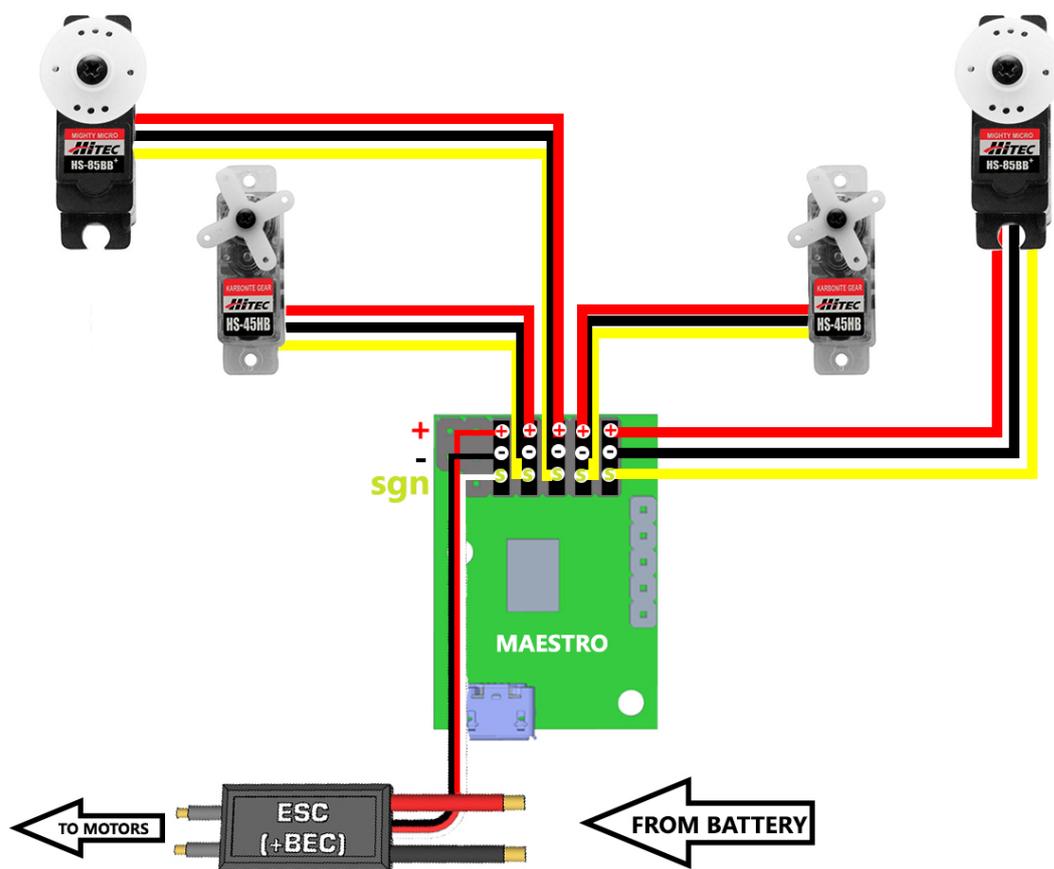


Figura 3.4: Il collegamento schematico tra il Maestro, i servomeccanismi e il controllore elettronico di velocità. In questa figura non viene indicato il collegamento tra il Raspberry e il Maestro che avviene grazie a un cavo miniUSB/microUSB.

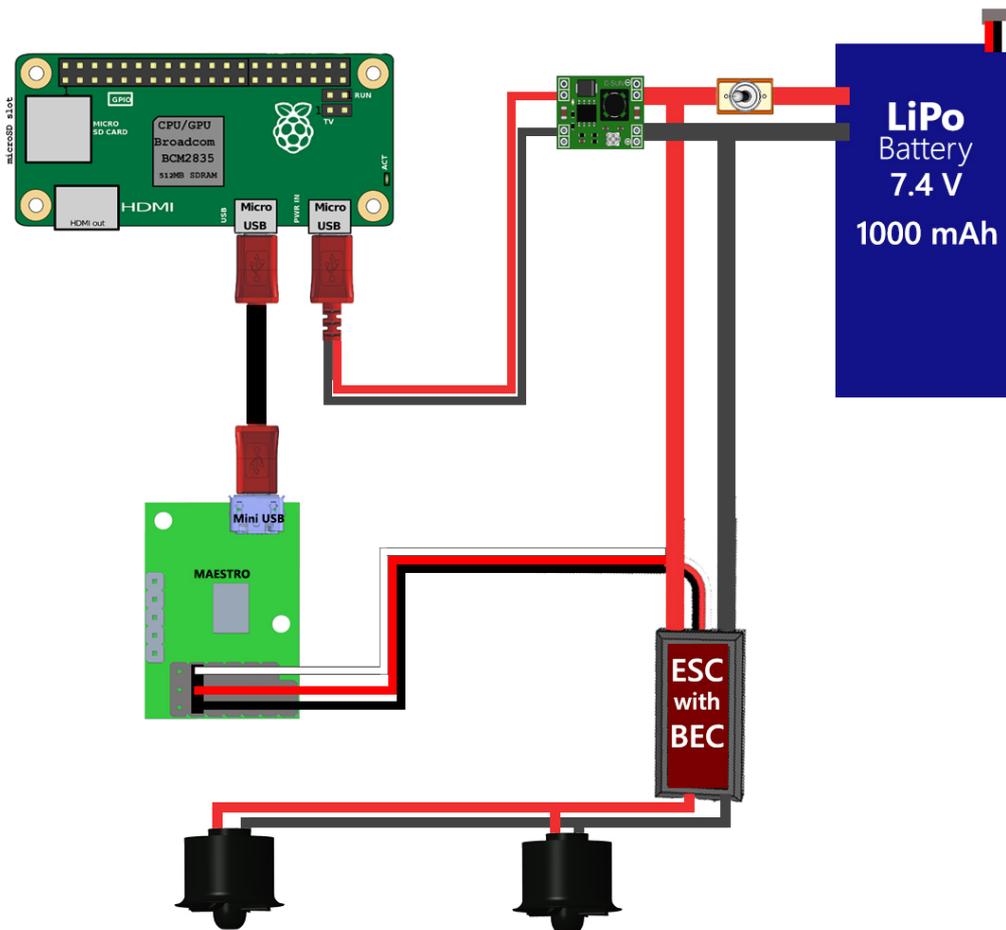


Figura 3.5: Lo schema di collegamento tra la batteria e i principali dispositivi che permettono la trasmissione dei dati: il Raspberry Pi e il Maestro. Tra quest'ultimo e l'on-board computer vi è presente un cavo miniUSB/microUSB che permette il passaggio dei dati. Questo cavo alimenta, inoltre, solamente l'elettronica di bordo del Maestro e non i servomeccanismi a esso collegati.

### 3.1.2.3 Collegamento: batteria, ESC e motori, Maestro e Raspberry

Tenendo in mente la figura 3.1, si schematizzano i dettagli del collegamento dei cavi di alimentazione nella figura 3.5. Il Raspberry è alimentato da un cavo USB creato ad hoc che preleva l'uscita dal regolatore di tensione a 5 V e, attraverso un adattatore, si inserisce nell'apposita entrata micro-USB dedicata all'alimentazione.

### 3.1.3 Durata della batteria

La nostra batteria, i cui dettagli sono stati proposti nella sezione dedicata e noti i consumi dei dispositivi elettrici, avrà una durata **in condizioni di massima scarica continua** di:

$$Batt. \text{ Duration } [min] = \frac{Cap. [mAh]}{Cap. [mAh] \cdot C \text{ rate} [A/h]} \cdot 60 = \frac{1000}{1000 \cdot 25} \cdot 60 = 2.4 [min]$$

Quindi la batteria<sup>2</sup>, messa in condizioni di massima scarica, cioè a 25 Ampere all'ora, essa si scaricherà completamente in giro di poco più di 2 minuti. Tuttavia, bisogna capire se c'è questa richiesta di corrente da parte degli utilizzatori. Poniamoci nel caso peggiore: tutti i dispositivi accesi e funzionanti a massimo regime. Dalla tabella 3.1 si sommano tutte le correnti assorbite in questione, osservando che il maggior utilizzo di corrente si ha per i servomeccanismi e soprattutto per le due ventole intubate, e si trova che per le due configurazioni considerate si hanno un consumo rispettivamente di 12102 mA per la prima e 13402 mA per la seconda configurazione. Mettendo questi numeri nella relazione che indica la durata in termini di minuti con tale richiesta di corrente, si ha, per la prima configurazione:

$$Batt\ Duration_1 [min] = \frac{Cap. [mAh]}{Absorbed\ current_1 [mA]} \cdot 60 = \frac{1000}{12102} \cdot 60 = 4.95 [min]$$

mentre per la seconda configurazione si ha:

$$Batt. Duration_2 [min] = \frac{Cap. [mAh]}{Absorbed\ current_2 [mA]} \cdot 60 = \frac{1000}{13402} \cdot 60 = 4.47 [min]$$

Queste condizioni sono poco realistiche, poiché non utilizzeremo mai tutti i servi contemporaneamente e in modo continuo. Per fare un'analisi più realistica bisogna tenere conto del tipo di utilizzo per ogni componente. Ad esempio, per i test che andremo a fare, le ventole vengono utilizzate per quasi tutta la durata della missione poiché esse servono a dare velocità al sistema e quindi al parafoil di avere portanza, mentre i servi che muovono le ventole saranno utilizzate in pochissime occasioni. I servi dedicati alla movimentazione dei fasci funicolari del parafoil anche essi verranno utilizzati saltuariamente e per poca durata, l'impiego necessario per manovrare la sonda nella direzione desiderata. Il resto del hardware è sempre acceso, ma hanno un impatto comunque basso in termini di potenza assorbita. Sommiamo tutte le correnti che vengono utilizzate in maniera continuativa per quasi tutta la durata della missione (vengono quindi esclusi i quattro servi). Si hanno 10384 mA di corrente assorbita.. Utilizzando la stessa formula, abbiamo che:

$$Batt. Duration_{no\ servo} [min] = \frac{Cap. [mAh]}{Absorbed\ current_{no\ servo} [mA]} \cdot 60 = \frac{1000}{10384} \cdot 60 = 5.77 [min]$$

Con un approccio conservativo possiamo dire che la batteria, in condizioni di massima spinta delle ventole durerà poco più di **5 minuti**. Nel capitolo dedicato ai test, vedremo se tali calcoli verranno confermati o meno.

## 3.2 Integrazione della struttura con l'elettronica di bordo

L'elettronica di bordo, si ricorda, è pressoché identica per entrambe le configurazioni. La differenza sta nella loro distribuzione sugli elementi strutturali e del loro peso. Li descriviamo in maniera separata poiché sono stati fatti in tempi diversi.

---

<sup>2</sup>Si ricorda che la batteria che utilizziamo si scaricherebbe in un'ora se il carico richiesto fosse di un Ampere. La capacità di 1000 mAh significa proprio questo.

### 3.2.1 Configurazione 1

In primo momento si incollano le componenti stampate intorno al contenitore centrale. Dopo aver lasciato così la struttura asciugare per ventiquattro ore, si incollano, si legano o si ancorano le componenti elettroniche a tale struttura, in dipendenza dalla possibilità di utilizzare una modalità o un'altra. Infine si collegano i cavi come descritto sopra.

Sull'anello superiore si incollano il GPS e la piattaforma inerziale, quest'ultima orientata in modo da avere l'asse longitudinale parallelo con la direzione di "avanti" del sistema e solidale con il sistema. Sempre su questa struttura, si incollano anche i servomeccanismi, assicurandoli ulteriormente con delle fascette in Nylon. I servo che devono manovrare il parafoil vengono muniti di due bracci forati in modo da poter legare ad essi le linee di comando. Il collegamento delle ventole ai loro servomeccanismi (figura 3.6 (b)) avviene per mezzo di un supporto ausiliario stampato in 3D, che, ha da una parte, la forma di un arco di diametro interno uguale al quello esterno della ventola, mentre dall'altra presenta un'asola della dimensione della leva del servomeccanismo, così da poter essere alloggiata in questo incavo. La ventola viene incollata al supporto ad arco su tutta l'area a disposizione che ha la larghezza quasi della ventola stessa. In secondo momento si avvita il fan con il supporto incollato al servomeccanismo, utilizzando l'incavo nella struttura per inserirvi dentro la leva del servomeccanismo, riuscendo in questo modo di movimentare la ventola intubata senza troppi sforzi sulla leva. La vista superiore si ha nella figura 3.6 (a).

Per la parte riguardante la batteria, il regolatore di tensione e il controllore della velocità e il loro supporto si noti la figura 3.7. Il supporto della batteria ha una struttura simmetrica: da entrambi i lati spuntano due lembi un po' più lunghi dedicati all'ancoraggio dell'ESC: infatti, si è pensato di utilizzarli entrambi per due ESC separati, ma in un secondo momento, si è deciso di utilizzare uno solo, poiché sufficiente.

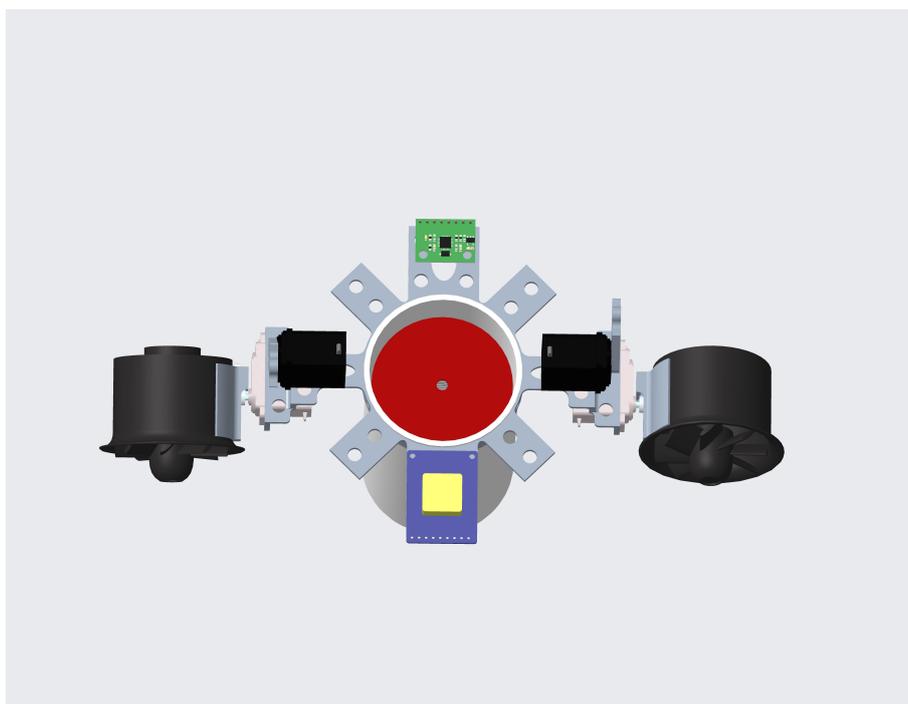
Nella parte diametralmente opposta al contenitore descritto sopra si trova la struttura di sostegno del computer, il Raspberry Pi stesso e il Maestro (figura 3.8).

#### 3.2.1.1 Protezione aggiuntiva per la Configurazione 1

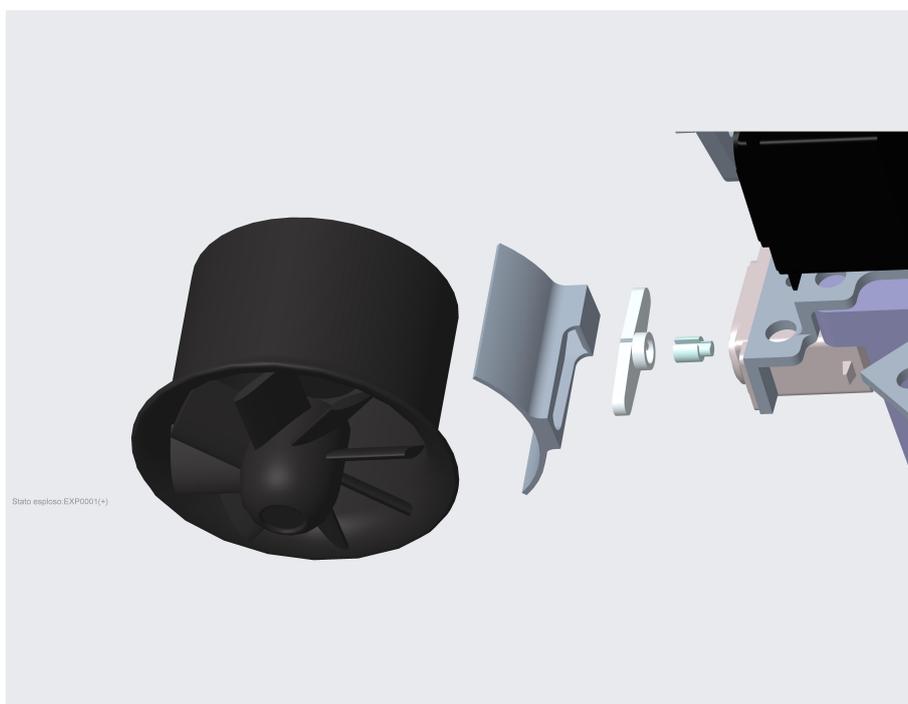
Dopo aver fatto dei test di resistenza alla flessione sulla parte di alloggiamento dei servi sull'anello superiore, è stato necessario introdurre una contro-ventatura per ridurre tale spostamento (vedasi figura 2.11), poiché la zona interessata è di massima sollecitazione sia dal punto di vista dell'intensità che di ripetizione: una rottura di tale collegamento sarebbe fatale.

In un secondo luogo, all'aggiunta della contro-ventatura si è annessa, con del nastro adesivo, una struttura in polistirolo con lo scopo di irrobustire il sistema e proteggerlo durante l'atterraggio. Nella figura 3.9 vengono indicate queste aggiunte utili all'irrobustimento e alla protezione della struttura, a scapito di un incremento del peso complessivo.

**Aerofreni** Un elemento di grande importanza del progetto, considerato anche nei lavori precedenti, è l'aerofreno. Lo scopo principale dell'aerofreno è l'aumento della resistenza aerodinamica durante la discesa e quindi di diminuire la velocità con cui

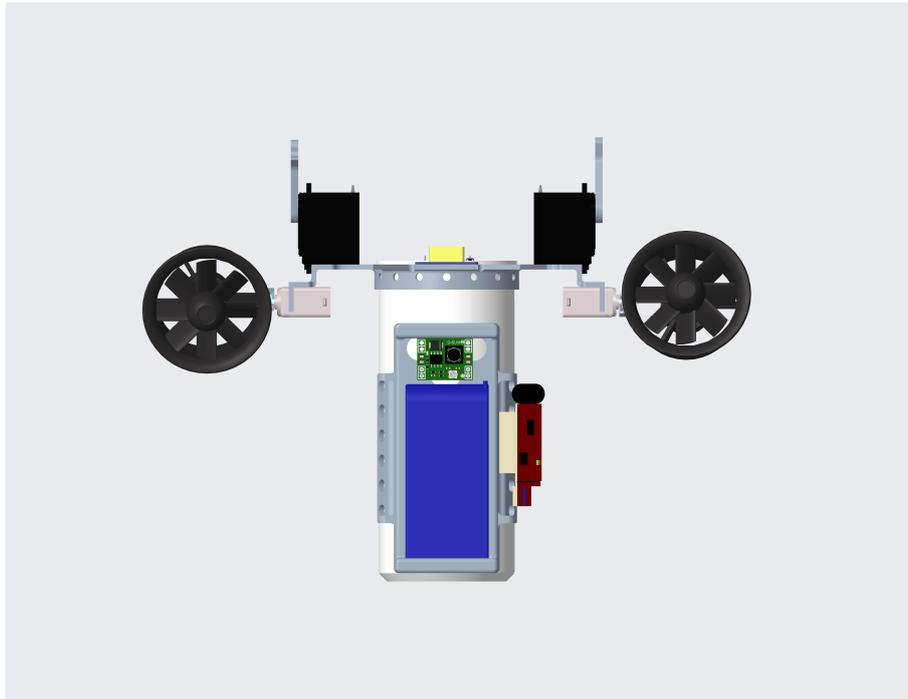


(a)

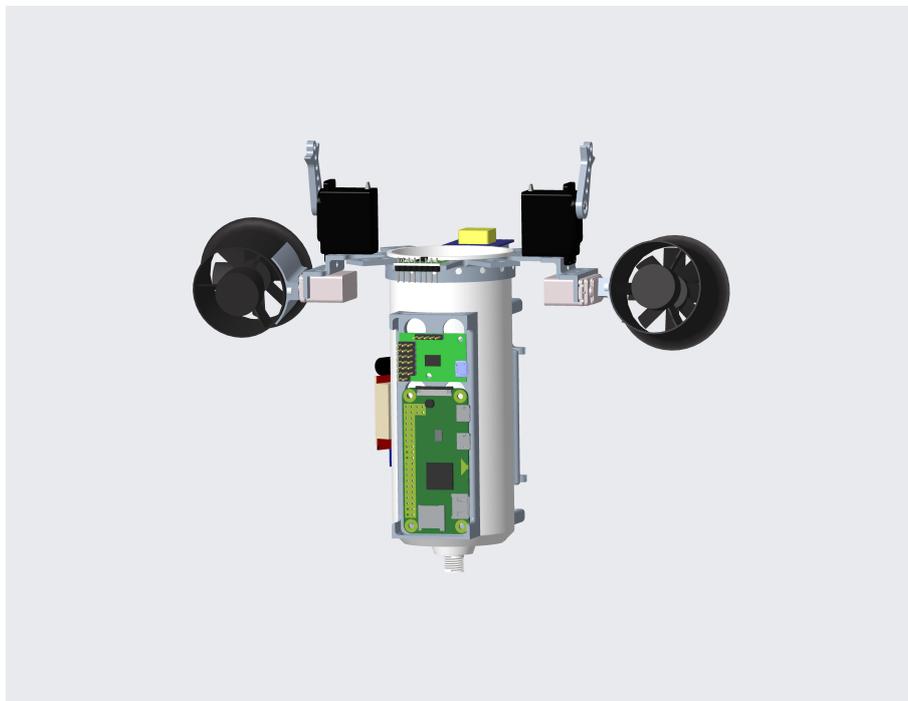


(b)

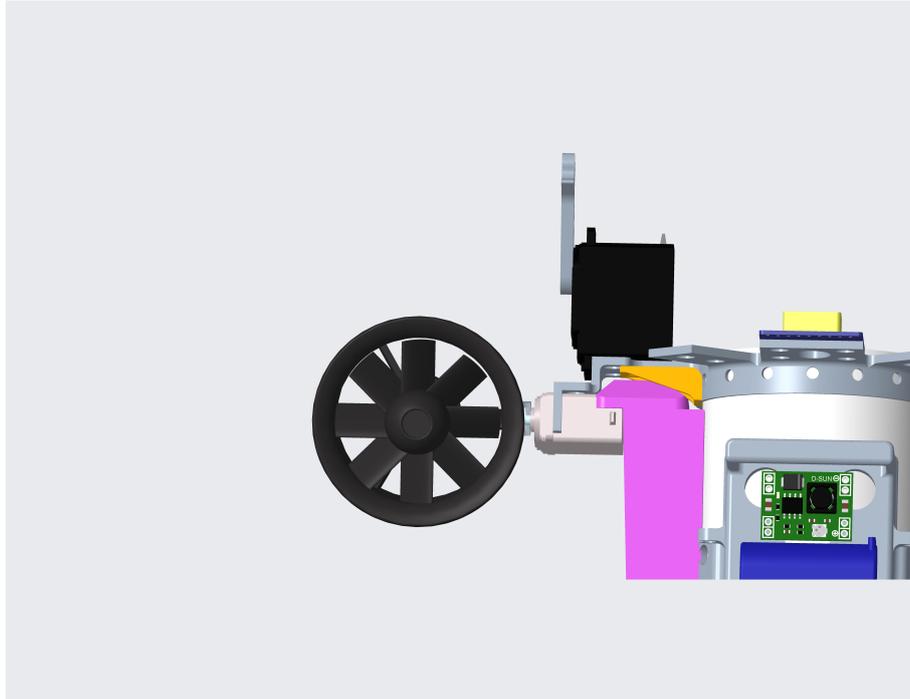
*Figura 3.6: (a) Vista da sopra della prima configurazione: si notano il GPS (blu) e la piattaforma inerziale (verde) ancorate alla struttura superiore. Ai lati si discernono i servomeccanismi utilizzati per la manovra del parafoil e le ventole collegate ai propri servomeccanismi, nel disegno di colore chiaro. (b) Vista esplosa del meccanismo di collegamento tra la ventola e il servo dedicato alla movimentazione.*



*Figura 3.7: Vista frontale della prima configurazione con in vista la batteria (blu) e del suo supporto stampato in 3D. Si nota anche il regolatore di tensione (verde), che serve per alimentare il Raspberry (qui non indicato) con il voltaggio giusto. Infine si nota di lato alla batteria anche il controllore elettronico di velocità (ESC: Electronic Speed Control) in rosso.*



*Figura 3.8: Prima configurazione: Vista dell'alloggiamento del Raspberry Pi e il Maestro, con l'apposito supporto.*



*Figura 3.9: Prima configurazione: Aggiunta di struttura di irrobustimento per diminuire la flessione del collegamento tra i servi e i fan con il corpo centrale (contro-ventatura in giallo). Aggiunta di struttura in polistirolo per irrobustire e proteggere l'intero sistema durante la fase di atterraggio (struttura in rosa scuro).*

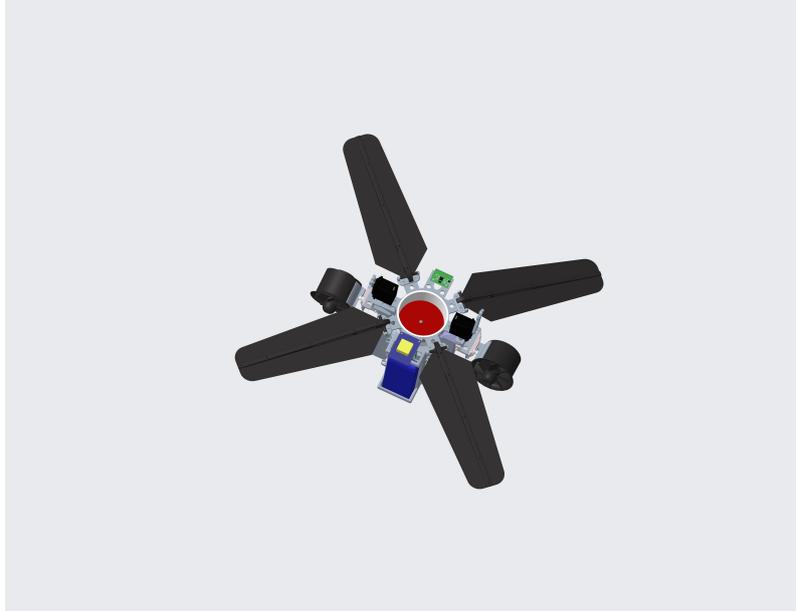
la sonda andrà a impattare a suolo. Vi è, tuttavia, una seconda grande utilità: la protezione del hardware durante la caduta o il ribaltamento. Infatti, durante i test, si sono dimostrati di grande utilità protettiva quando si andava a impattare contro ostacoli in direzione orizzontale, come ad esempio muri. Durante l'impatto, sono gli aerofreni, che sporgono dalla struttura in direzione orizzontale in tutte le direzioni, ad assorbire il colpo, per poi flettersi: l'aggiunta di questi ultimi, infatti, deve comportare il minimo aumento di peso. Nel disegno CAD si è utilizzato un aerofreno progettato nei lavori precedenti, poiché l'idea era di riutilizzarli. Essi sono stampati in 3D e hanno una struttura rigida. Nella realtà, a causa del peso eccessivo di questi ultimi, abbiamo riprogettato tali aerofreni con materiali molto più leggeri, ma anche flessibili: si sono utilizzate delle cannucce per la superficie e del fil di acciaio per la rigidità. Nella figura 3.10 viene indicato il loro collocamento sia nella versione originale (a) che nella nuova versione (b).

Infine, nella figura 3.11 viene indicato la prima configurazione al completo.

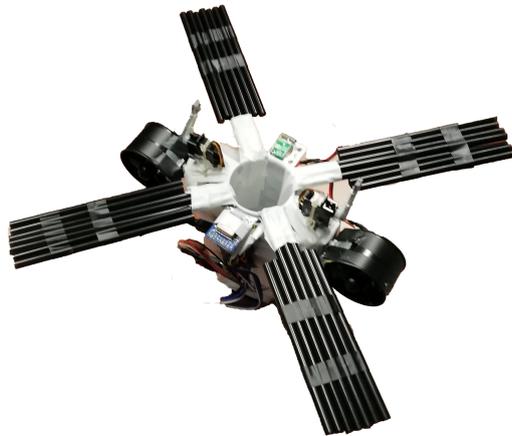
### 3.2.2 Configurazione 2

La seconda configurazione, come accennato nella parte sulla costruzione dei suoi elementi strutturali, viene integrata in maniera diversa: ora il fulcro di incollaggio non è più il contenitore del parafoil (qui completamente mancante), bensì la struttura superiore a cui in seguito verrà incollato il corpo in polistirolo, contenente la batteria, l'ESC e il Raspberry.

Alla struttura superiore per prima cosa si incollano le strutture ausiliarie che devono trattenere i servomeccanismi. Di seguito, si incollano i servi nella loro posizione,

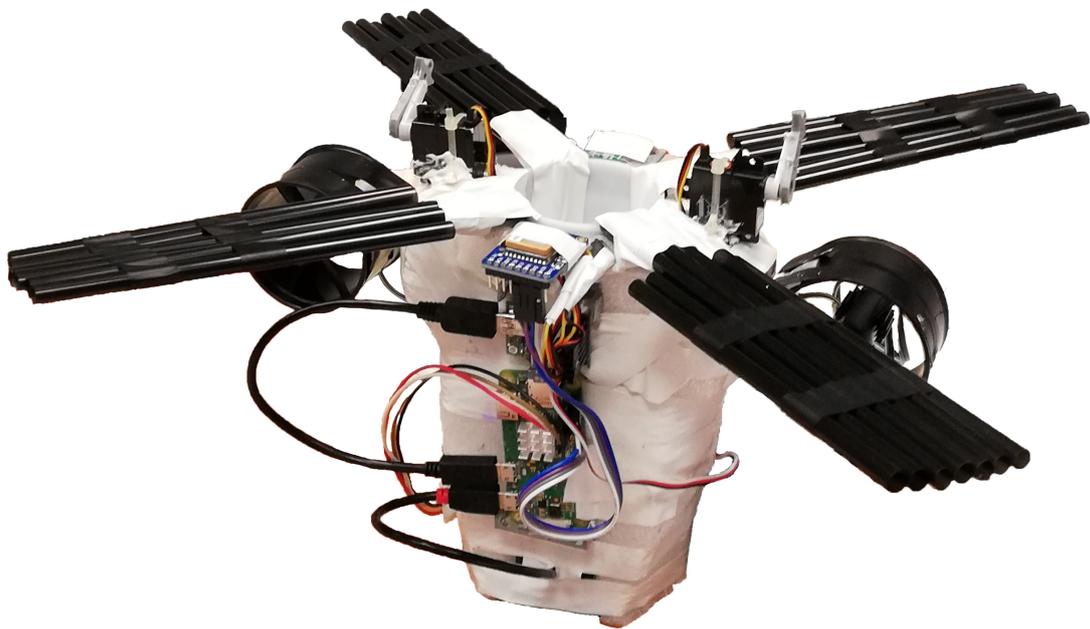


(a)

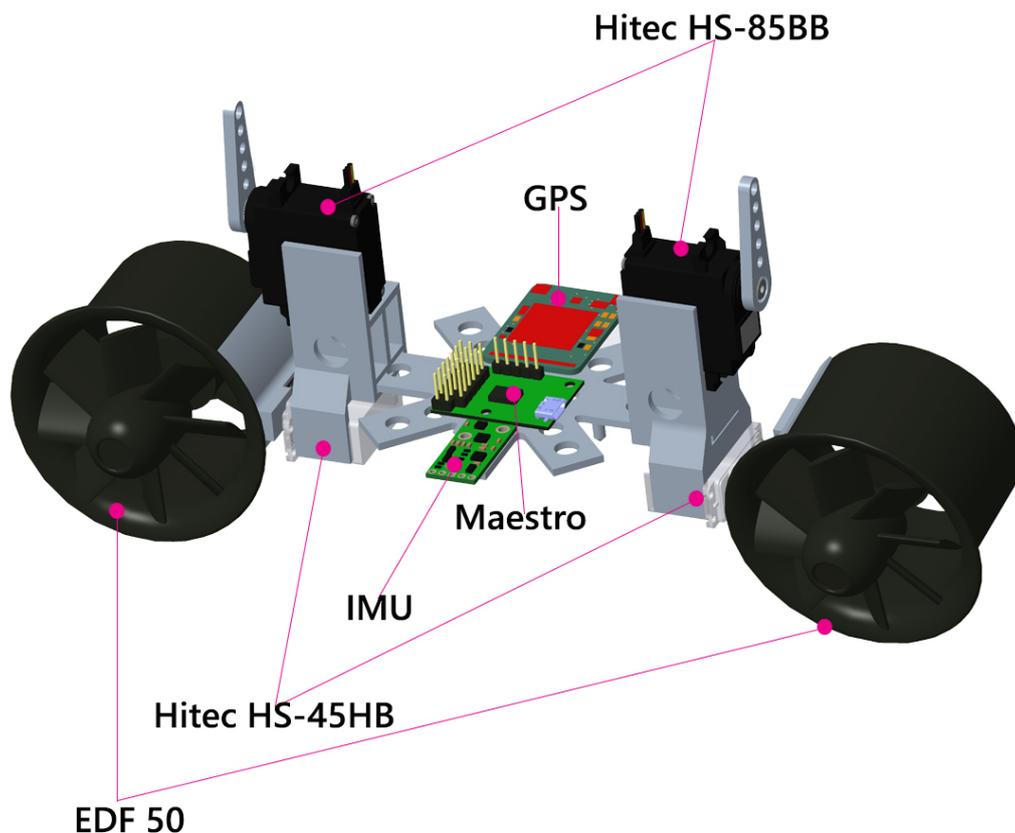


(b)

*Figura 3.10: Gli aerofreni della prima configurazione: (a) quelli progettati precedentemente: rigidi, stampati in 3D, ma molto più pesanti dei requisiti; (b) progettati ex novo: un insieme di cannuce posizionate a creare una superficie utile per la creazione della resistenza aerodinamica e per attutire l'impatto con ostacoli in direzione orizzontale o in caso di ribaltamento. Il fil di ferro fa da guida dentro ogni aerofreno e ha anche il compito di irrobustire la struttura.*



*Figura 3.11: La prima configurazione completata, a meno del parafoil. Le strutture di supporto in polistirolo presenti sui lati rendono la struttura molto più voluminosa, ma più robusta. Da notare la presenza del cablaggio, non indicato nei disegni CAD: Spunta all'occhio il cavo miniUSB/microUSB che collega il RPi al Maestro, i cavi che connettono le unità GPS e l'IMU (a sinistra) e i cavi provenienti dall'ESC (sulla destra). Il cavo dell'alimentazione viene dal basso dove è stato collocato un interruttore.*

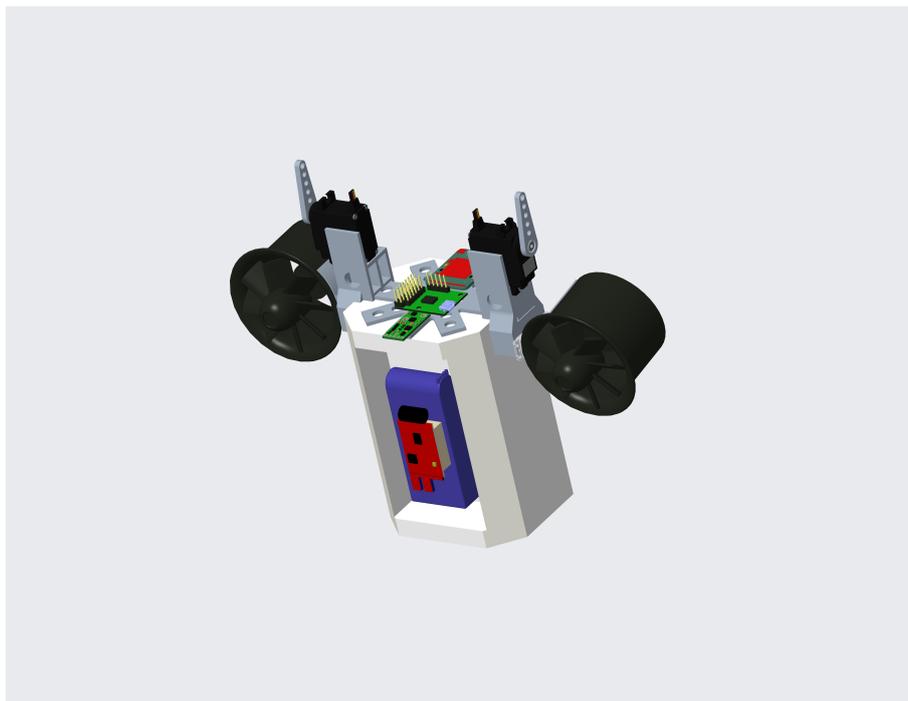


*Figura 3.12: Seconda configurazione. La struttura superiore, a differenza della prima, alloggia anche il Maestro. Le strutture ausiliarie, che alloggiano i servi, sono una conseguenza del fatto che nella prima configurazione qui si trovava il punto più debole della struttura, sperimentando tale risultato in diverse occasioni. Così progettato, si pensa di poter migliorare la robustezza della struttura, senza inficiarne la sua leggerezza complessiva.*

mentre il modulo GPS, l'IMU e il Maestro vengono incollati centralmente alla struttura. Il disegno nella figura 3.12 mostra come questi elementi hardware sono posizionati. La maggiore differenza sta nel fatto che il Maestro è posizionato sopra, dandoci la possibilità di risparmiare peso sul cablaggio tra i servo e il Maestro stesso, ora che non abbiamo più il contenitore centrale. In effetti, ora il parafoil deve essere portato esternamente. Non cambia la logica di collegamento tra le ventole e l'albero del servomeccanismo. Viene ristampata la struttura di collegamento ad arco, con delle piccole modifiche, in modo da irrobustirla.

Per l'alloggiamento del hardware restante si è deciso, sin da subito, di utilizzare una struttura in polistirolo, facilmente lavorabile. Il collegamento tra il corpo e la struttura superiore si ha attraverso l'incollaggio dei denti (vedasi figura 2.19 (b)) della stessa struttura nei fori creati apposti nel corpo in polistirolo. L'incollatura viene ulteriormente evidenziata con l'applicazione di un paio di cerchietti di fil di ferro tra i fori della struttura superiore e il corpo.

I dispositivi elettronici rimanenti vengono suddivisi su due scanalature: da una parte viene messa la batteria con l'ESC, mentre dall'altra vengono alloggiati il RPi, il regolatore di tensione e l'interruttore.



(a)

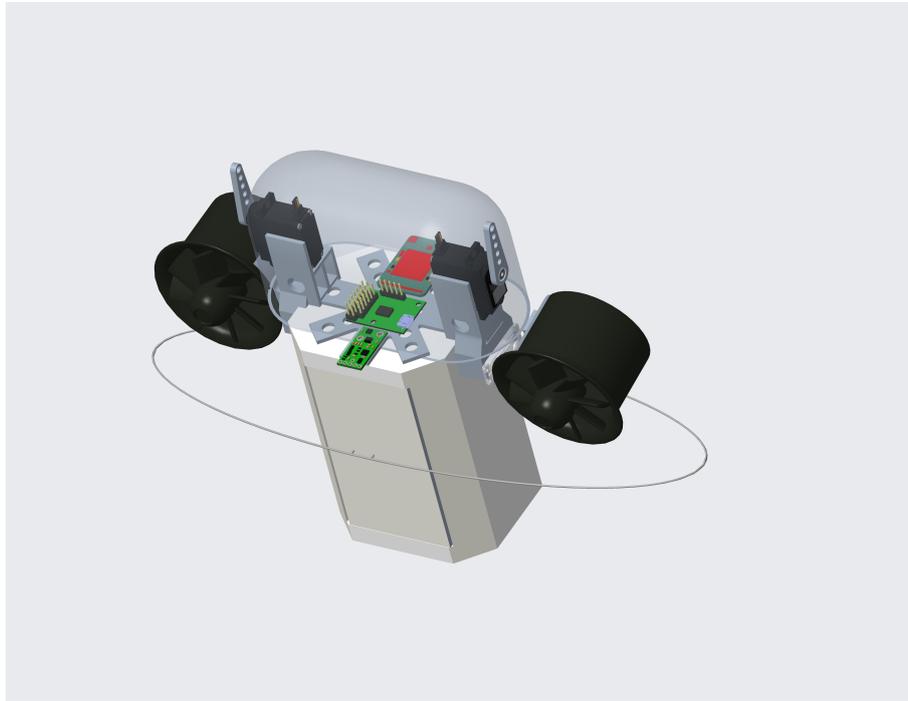


(b)

*Figura 3.13: Seconda configurazione: il hardware sui due lati del corpo in polistirolo: (a) batteria e il controllore elettronico di velocità con BEC; (b) Raspberry Pi Zero, regolatore di tensione step-down e l'interruttore.*

### 3.2.2.1 Protezione aggiuntiva per la Configurazione 2

Le strutture protettive per questa seconda configurazione sono diverse. Prima di tutto si creano due pareti, sempre di polistirolo, per coprire l'elettronica di bordo esposta. In seguito si crea una struttura a forma ellittica, fatta di fil di ferro, che ha la funzione di proteggere sia le ventole sia, quindi, anche il collegamento con i servi in caso di caduta laterale. Questa protezione viene collegata con il corpo del sistema in maniera rigida e ha la funzione degli aerofreni nella prima configurazione, nel caso di cadute in direzione laterale e impatti indesiderati con i fan. Infine si aggiunge una cupola in plastica trasparente nella parte superiore con la funzione di proteggere l'elettronica posizionata nella parte superiore della struttura. Nella figura 3.14 si visualizza il disegno CAD completo della seconda configurazione (a) e l'immagine reale (b).



(a)



(b)

*Figura 3.14: Seconda configurazione: (a) disegno CAD della seconda configurazione con le protezioni. Mancano i cablaggi e il Parafoil. In questa configurazione, rispetto alla precedente, si hanno meno cavi esposti: quasi tutti i cavi sono inglobati nella struttura o dalle protezioni aggiuntive; (b) immagine reale della seconda configurazione.*

## 4 | Impostazione dei software

I vari dispositivi devono comunicare tra di loro e quindi vi è la necessità di installare alcuni programmi e librerie apposite per permettere il corretto funzionamento di tale comunicazione, dopo aver installato prima di tutto un sistema operativo. Di seguito vedremo la logica e gli strumenti utilizzati per ognuno dei componenti e per il sistema nel suo complesso.

Alla fine del capitolo vedremo le relazioni tra i programmi che gestiscono i dati di ogni dispositivo, in input e in output, inseriti dentro un programma centrale, il cui file sorgente è stato scritto in C.

### 4.1 Raspberry Pi Zero W

**Sistema Operativo (SO)** Lo sviluppatore degli on-board computer *Raspberry Pi* ha predisposto un *SO* appartenente alla famiglia di sistemi operativi *Debian GNU/Linux*, quindi a licenza libera e supportante piattaforme di tipo *ARM*<sup>1</sup>, utilizzate in sistemi integrati come il nostro appunto: **Raspbian**.

**Installazione** L'installazione del *SO* avviene su una scheda di memoria di tipo micro-SD da 16 GB, per la quale si ha uno slot dedicato sul dispositivo hardware a nostra disposizione. Il sistema operativo Raspbian non viene installato direttamente, ma viene utilizzato un *installer* di nome **NOOBS**, acronimo di *New Out Of the Box Software*, che contiene Raspbian.

Prima di procedere con l'installazione assicurarsi di avere uno schermo funzionante, un cavo HDMI che possa essere inserito nel Raspberry Pi e un mouse con cavo microUSB che possa essere utilizzato una volta acceso il computer. Nota: vi è una possibilità di installare e configurare il sistema operativo senza la necessità di utilizzare uno schermo, modalità qui non riportata.

L'installazione avviene con i seguenti passaggi:

1. Scaricare la cartella contenente NOOBS in formato `.zip` dal sito ufficiale del Raspberry Pi [19] sul proprio PC. Il file `.zip` è grande circa 1.5 GB.
2. Inserire la scheda microSD da 8 GB o più nel PC.
3. Formattare la scheda con i seguenti passaggi (per Windows solo):
  - Scaricare il tool di formattazione della SD Association dal sito ufficiale [23] o uno alternativo

---

<sup>1</sup>ARM, precedentemente noto come Advanced RISC Machine, è una famiglia di microprocessori relativamente semplici a 32-bit utilizzata nei sistemi di tipo embedded (come nel caso di un utilizzo di un Raspberry Pi).

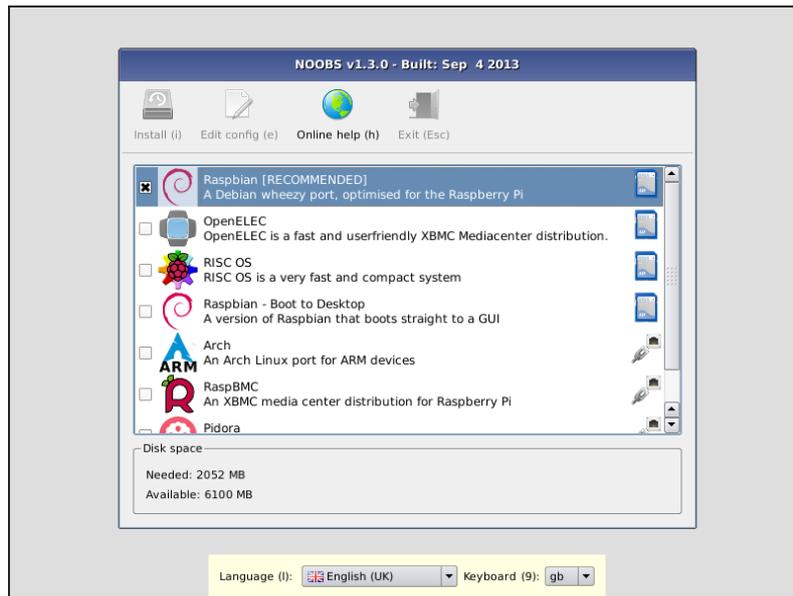


Figura 4.1: Esempio di schermata prima di installare Raspbian sul Raspberry Pi.

- Installare ed eseguire il tool di formattazione sul proprio PC
  - Controllare che sia selezionata la scheda micro-SD
  - Premere il tasto "Format"
4. Estrarre i file contenuti nel file NOOBS\_v2\_8\_2.zip (nel nostro caso è la versione 2.8.2)
  5. Copiare i file appena estratti nella cartella principale (root directory) della scheda micro-SD appena formattata, quindi espellere in maniera sicura la scheda.
  6. Inserire la micro-SD nel Raspberry Pi e connettere l'alimentazione.

Connessi precedentemente il cavo HDMI allo schermo e il mouse alla porta USB disponibile, si accende una schermata iniziale come nella figura 4.1 quando viene portata l'alimentazione alla porta USB dedicata. Il Raspberry Pi Zero W ha una rete Wi-Fi integrata, quindi per default cerca di connettersi a reti disponibili: creare quindi una connessione a una rete Wi-Fi. Alla schermata che compare scegliere di installare Raspbian e premere "Install".

**Configurazioni Iniziali** Un punto a sfavore del RPi Zero W, a causa della sua leggerezza, è che possiede solo una periferica micro-USB disponibile. Sarà necessario utilizzare sia il mouse e la tastiera inizialmente, solo per le configurazioni iniziali, e ciò implica trovare una soluzione che ci lasci utilizzare entrambe le periferiche. Finita l'installazione del *SO*, è buona norma (per motivi di sicurezza) cambiare il nome e la password del dispositivo. Vi è anche la necessità di cambiare alcune configurazioni per permettere l'utilizzo del Raspberry in remoto. Bisogna attivare l'opzione SSH nel menu della configurazione. SSH (**S**ecure **S**hell) è un protocollo di interfaccia di comunicazione che stabilisce una sessione remota con un altro *host* di una stessa rete

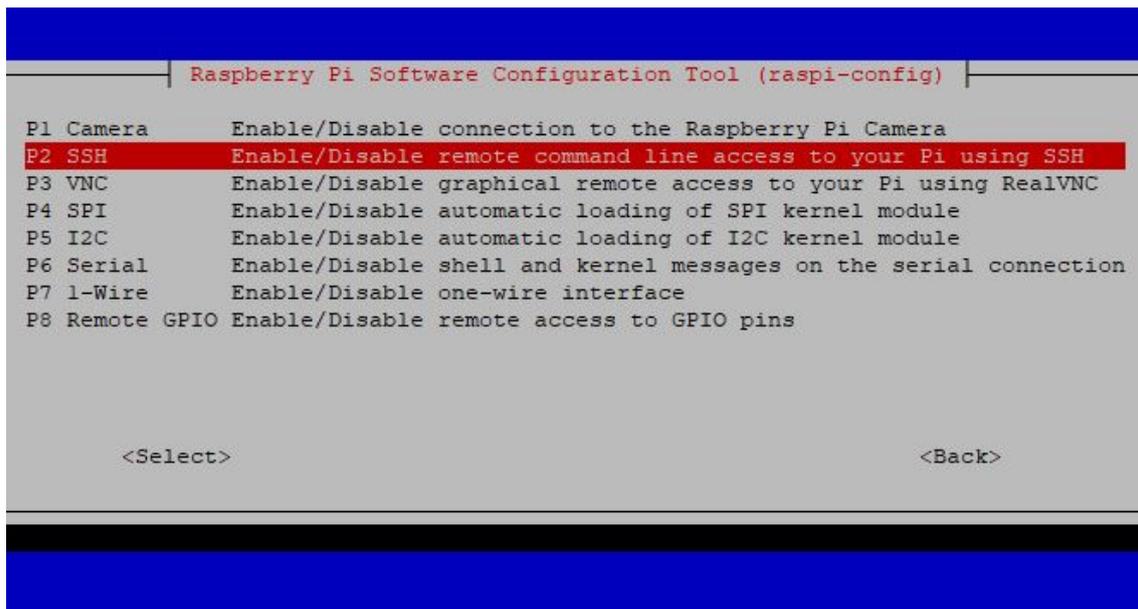


Figura 4.2: Attivazione delle periferiche (5 Interfacing Options) dentro il tool di configurazione eseguito con il comando `sudo raspi-config`. Si eseguono tutte le modifiche necessarie per poter lavorare in remoto (attivazione di SSH) e per comunicare con altre interfacce come I2C per la piattaforma inerziale.

informatica. Ci sono due modalità per fare queste modifiche: o si utilizza il mouse e lo schermo e si cerca nel menu a tendina la configurazione e si eseguono le modifiche oppure si utilizza la linea di comando attraverso il terminale del Raspbian. Abbiamo utilizzato questa seconda possibilità e per fare le modifiche bisogna eseguire, nel caso di attivazione di SSH:

1. Aprire il terminale con il comando `ctrl+t`
2. Scrivere `sudo raspi-config`
3. Selezionare Interfacing Options
4. Scendere fino a SSH, selezionarlo premendo Invio
5. Scegliere Yes
6. Selezionare Ok
7. Scegliere Finish

Lo stesso ragionamento va fatto anche per le altre periferiche che andremo a utilizzare. Da attivare, oltre al SSH, sono I2C per la piattaforma inerziale, Serial per il GPS e Remote GPIO. Altre attivazioni sono opzionali, come ad esempio Camera che permette di utilizzare l'uscita dedicata al collegamento con un dispositivo per acquisizione di immagini. Nella figura 4.2 viene mostrata la schermata a cui si deve arrivare per fare le modifiche richieste.

È preferibile utilizzare una stessa rete Wi-Fi, così da evitare di inserire le credenziali per accesso all'internet tutte le volte che si cambia la rete. Nel nostro caso abbiamo utilizzato un cellulare come hotspot Wi-Fi.

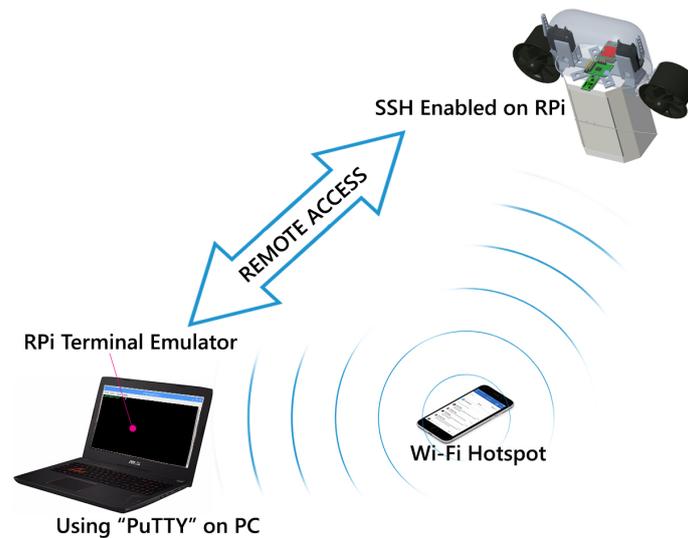


Figura 4.3: Schema semplificato della connessione in remoto: attivare SSH sul Raspberry, connettersi alla stessa rete Wi-Fi, utilizzare PuTTY per emulare il terminale del Raspbian installato precedentemente.

**Connessione in remoto** *PuTTY* è un software libero che emula il terminale di un sistema informatico, dando la possibilità quindi di gestire tale sistema in remoto. Supporta diversi protocolli di connessione, come il già introdotto SSH, presente sul Raspberry Pi. Si ricorda che i due dispositivi che dovranno comunicare tra di loro (il PC e il Raspberry) dovranno essere connessi alla stessa rete Wi-Fi (nel nostro caso utilizziamo l’hotspot di un cellulare). Nella figura 4.3 viene indicato lo schema semplificato di come funziona il collegamento in remoto. L’installazione e l’esecuzione sono:

1. Scaricare `putty.exe` dal sito ufficiale [22]. Il `putty.exe` è sufficiente se viene utilizzato il protocollo di tipo SSH.
2. (Opzionale) creare una connessione ad accesso rapido sul proprio Desktop o dove è più comodo.
3. Cliccare due volte sull’eseguibile `putty.exe`. Compare la schermata indicata nella figura 4.4.
4. Nella casella Host Name (or IP address<sup>2</sup>) inserire il nome del dispositivo a cui si vuole collegare o il suo IP (nel nostro caso l’IP del Raspberry Pi). Lasciare la Port a 22, come da default.
5. Lasciare il tipo di connessione su SSH.
6. Premere `Open` per iniziare una sessione SSH.
7. In caso di prima connessione compare un messaggio di avviso di sicurezza, premere `Ok`.

<sup>2</sup>IP adress sta per Internet Protocol address e individua univocamente, attraverso un numero identificativo, un dispositivo collegato a una rete informatica, che utilizza l’Internet Protocol come modalità di interazione in rete.

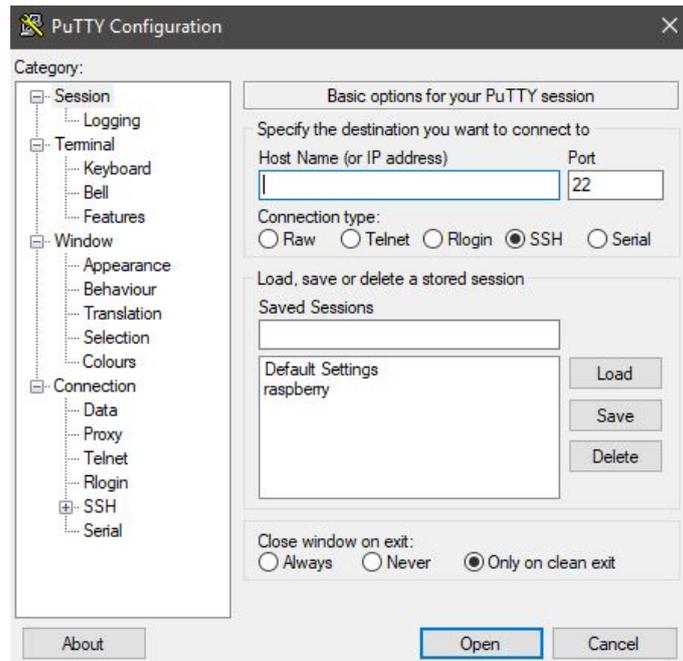


Figura 4.4: Schermata iniziale di PuTTY.

8. Una volta aperta la connessione SSH, compare il prompt dei comandi (terminale) che richiede il login e la password del sistema a cui vogliamo connetterci (sono quelli impostati inizialmente nelle configurazioni iniziali).
9. Inserite le credenziali, è possibile cominciare a utilizzare il terminale normalmente

Gli indirizzi IP sono visibili dal dispositivo che ha il ruolo di Wi-Fi Hotspot, nel nostro caso il cellulare. Essi vanno inseriti dentro PuTTY per creare la connessione in remoto.

## 4.2 Attivazione del GPIO

Il GPIO (General Purpose Input/Output) è un'interfaccia fisica del Raspberry, descritta dalla presenza dei pin, che permette il collegamento con dispositivi o periferiche esterne, nel nostro caso la connessione con il GPS e la piattaforma inerziale. Nel capitolo dedicato all'integrazione del hardware abbiamo schematizzato il collegamento fisico tra il GPIO del Raspberry e le periferiche esterne. Per farlo funzionare, serve tuttavia attivarlo. L'attivazione avviene con l'installazione di una libreria chiamata **WiringPi**, che è una libreria di accesso GPIO basata su PIN scritta in C per dispositivi con *SoC BCM2835*, quindi quello utilizzato sul nostro Raspberry Pi Zero. Sul sito ufficiale della WiringPi [40] viene indicato come installarla sul proprio dispositivo (seguire tale procedimento se si vuole installare la libreria). Il corretto funzionamento è garantito per piattaforme di tipo Raspberry Pi. L'installazione avviene attraverso la riga di comando tramite il git-core. A installazione finita, possiamo controllare l'utilizzo dei pin in input e output con il comando `gpio readall`, la cui risposta viene indicata nella figura 4.5, con tutti i dettagli.

Per le nostre applicazioni ci limitiamo ad utilizzare solo i primi dieci pin, che ci

```
pi@raspberrypi:~$ gpio readall
```

Pi ZeroW											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
		3.3v			1	2		5v			
2	8	SDA.1	ALTO	1	3	4		5v			
3	9	SCL.1	ALTO	1	5	6		0v			
4	7	GPIO. 7	IN	0	7	8	1	ALT5	TxD	15	14
		0v			9	10	1	ALT5	RxD	16	15
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13	14			0v		
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4	23
		3.3v			17	18	0	IN	GPIO. 5	5	24
10	12	MOSI	ALTO	0	19	20			0v		
9	13	MISO	ALTO	0	21	22	0	IN	GPIO. 6	6	25
11	14	SCLK	ALTO	0	23	24	1	OUT	CE0	10	8
		0v			25	26	1	OUT	CE1	11	7
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31	1
5	21	GPIO.21	IN	1	29	30			0v		
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33	34			0v		
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	20
		0v			39	40	0	IN	GPIO.29	29	21

Figura 4.5: Schermata dei pin del GPIO. Per ottenere tale tabella eseguire il comando `gpio readall` dal terminale. Il riquadro rosso indica i nomi fisici dei pin. I riquadri blu indicano i nomi funzionali dei pin. Le colonne intestate con 'V' indicano i livelli di voltaggio dei pin: non sono presenti numeri su quelli che non sono pin di tipo GPIO (alimentazioni e messe a terra). Lo zero indica che il voltaggio è basso (0 Volt), l'uno indica che il voltaggio è alto (3.3 V). Le colonne intestate con 'Mode' indicano come i pin possono essere utilizzati: se *in*, pin usato come input; se *out*, pin usato come output; se *ALT[#]* pin usato con una funzione speciale. La colonna 'wPi' è simile alla colonna 'Name' senza il nome "GPIO". Infine, la colonna 'BCM' indica il numero di periferiche AMR, non di nostro interesse.

permetteranno di comunicare con la piattaforma inerziale e l'unità GPS.

### 4.3 Attivazione dell'unità GNSS

Il sito ufficiale dell'Adafruit fornisce un tutorial<sup>[3]</sup> su come impostare la comunicazione tra il dispositivo e un Raspberry Pi.

Affinché il Raspberry Pi Zero legga correttamente i dati seriali ricevuti attraverso il collegamento *UART* con il dispositivo GPS, è necessario installare un programma demone (*daemon*), chiamato **gpsd**<sup>[9]</sup>. Esso è un daemon di servizio che monitora uno o più ricevitori GPS collegati a un computer host tramite porte seriali o USB, rendendo tutti i dati sulla posizione, rotta e velocità dei sensori disponibili per essere interrogati sulla porta TCP 2947<sup>3</sup> del computer host. Inoltre, **gpsd** risponde alle query (interrogazioni) con un formato che è sostanzialmente più semplice da analizzare rispetto a NMEA-0183<sup>4</sup>, emesso dalla maggior parte dei GPS. Infatti, il formato delle sentences inviate dal *talker* hanno un formato standard lungo fino a massimo 82 caratteri, un esempio di output NMEA-0183 viene indicato di seguito:

```
$GPGGA,092750.000,5321.6802,N,00630.3372,W,1,8,1.03,61.7,M,55.2,M,,*76
$GPGSA,A,3,10,07,05,02,29,04,08,13,,,,,1.72,1.03,1.38*0A
$GPGSV,3,1,11,10,63,137,17,07,61,098,15,05,59,290,20,08,54,157,30*70
$GPGSV,3,2,11,02,39,223,19,13,28,070,17,26,23,252,,04,14,186,14*79
$GPGSV,3,3,11,29,09,301,24,16,09,020,,36,,,*76
$GPRMC,092750.000,A,5321.6802,N,00630.3372,W,0.02,31.66,280511,,,A*43
$GPGGA,092751.000,5321.6802,N,00630.3371,W,1,8,1.03,61.7,M,55.3,M,,*75
$GPGSA,A,3,10,07,05,02,29,04,08,13,,,,,1.72,1.03,1.38*0A
$GPGSV,3,1,11,10,63,137,17,07,61,098,15,05,59,290,20,08,54,157,30*70
$GPGSV,3,2,11,02,39,223,16,13,28,070,17,26,23,252,,04,14,186,15*77
$GPGSV,3,3,11,29,09,301,24,16,09,020,,36,,,*76
$GPRMC,092751.000,A,5321.6802,N,00630.3371,W,0.06,31.66,280511,,,A*45
```

La loro cifratura non è complicata e fornisce molte informazioni utili come ad esempio che tipo di costellazione di satelliti viene interrogato. Ad esempio *\$GPGSV* sta per *GPS Satellites in view*. I dettagli delle sigle e l'interpretazione delle *sentences* possono essere lette sui siti dedicati <sup>[18]</sup>. Nel caso delle nostre applicazioni, si preferisce utilizzare il **gpsd** che ritorna dei risultati in formato più leggibili, in termini di latitudine, longitudine e altitudine.

Questo daemon gestisce con attenzione gli errori di analisi e fornisce un'interfaccia comune e ben definita a qualsiasi modulo GPS, quindi, agisce come una sorta di filtro tra i dati grezzi forniti dal hardware attuale del modulo e i dati utili pronti all'utilizzo.

---

<sup>3</sup>TCP (Transmission Control Protocol) è un protocollo di rete a pacchetto di livello di trasporto, appartenente alla suite di protocolli Internet, che si occupa del controllo di trasmissione ossia rende affidabile la comunicazione dati in rete tra mittente e destinatario.

<sup>4</sup>NMEA (National Marine Electronics Association) è un standard di comunicazione utilizzato nella trasmissione dei dati satellitari di tipo GPS. Il principio è che solo il talker (la fonte) può trasmettere dati (sentences) e la listener (la ricevente) può riceverli.

Il `gpsd` fornisce anche l'importante possibilità di utilizzarlo direttamente con i programmi scritti in linguaggio **C**, come nel nostro caso. Basta includere nel file sorgente `#include < gps.h >` e durante la costruzione dell'eseguibile utilizzare il linker `-lgps`. La sintassi di tale libreria - importante per l'estrazione delle coordinate - è indicata sempre dal sito ufficiale [13].

#### 4.3.0.1 Test sul funzionamento del `gpsd`

L'unità GNSS inizia a trasmettere correttamente dati quando ha una connessione stabile con un numero sufficienti di satelliti. Tale condizione viene chiamata di **fix**. L'Ultimate GPS Breakout V3 a nostra disposizione ha inglobato un LED di **fix** che ci fornisce l'informazione riguardante la connessione stabile o meno del modulo. Il LED lampeggia a circa 1 Hz mentre cerca i satelliti e una volta trovata la soluzione lo fa una volta ogni 15 secondi, in modo da risparmiare la batteria.

Nota importante: il ricevitore GPS deve trovarsi all'aperto, o comunque deve avere una buona visuale del cielo. La condizione di **fix** con il GPS al chiuso non sarà possibile.

A installazione finita di `gpsd`, per testare bisogna eseguire i seguenti comandi:

- `sudo killall gpsd`: "uccide" il processo di `gpsd`, ha la funzione di azzerare una connessione precedente di `gpsd`. Una sorta di *clear* totale.
- `sudo stty -F /dev/ttyS0 9600 raw 9600 cs8 clocal -cstopb`: crea la connessione con la porta seriale `ttyS0`.
- `sudo gpsd /dev/ttyS0 -F /var/run/gpsd.sock`: apertura del processo `gpsd` sulla porta seriale `ttyS0`.

La connessione è innescata. Al comando `cgps -s`<sup>5</sup> del programma `gpsd` si ha il risultato indicato nella figura 4.6.

## 4.4 Attivazione della piattaforma inerziale

L'attivazione dell'IMU avviene per mezzo della già installata libreria `WiringPi`. I sensori a bordo della piattaforma (una combinazione di un giroscopio L3GD20H e un accelerometro e un magnetometro LSM303D tutti 3-assiali) comunicano con il Raspberry Pi attraverso il bus I<sup>2</sup>C (Inter Integrated Circuit), descritto nel capitolo tre.

A partire dai datasheet [37, 38] dei sensori, da cui vengono acquisiti i registri degli indirizzi necessari alla comunicazione, si sviluppa un programma in **C** che possa leggere i dati trasmessi dai canali con i nomi dei registri noti, mediante l'ausilio delle funzioni di `WiringPiI2C` (la libreria è proprio pensata a facilitare l'utilizzo dell'interfaccia I<sup>2</sup>C del Raspberry Pi [41]) e dia in output i valori delle tre accelerazioni lineari rispetto alla Terra, delle tre velocità giroscopiche e dei tre campi magnetici. Combinati insieme attraverso un filtro, i cui dettagli vedremo in seguito, otterremo l'assetto della piattaforma in termini degli angoli di imbardata, beccheggio e rollio (*yaw*, *pitch*, *roll*). Il lavoro non viene svolto *ex novo*, ma riprende le funzioni di

---

<sup>5</sup>Vi è la possibilità di visualizzare più dettagli con il comando dalla linea di comando `gpsmon`.

```

lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqkllqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x   Time:      2018-08-15T14:34:57.000Z   xxPRN:   Elev:   Azim:   SNR:   Used:  x
x   Latitude:  45.      N                xx  18    80    027    33    Y    x
x   Longitude: 7.      E                xx   1    60    307    31    Y    x
x   Altitude:  211.8 m                 xx  22    59    241    39    Y    x
x   Speed:     0.2 kph                  xx   8    49    175    34    Y    x
x   Heading:   67.6 deg (true)          xx  14    44    098    25    Y    x
x   Climb:     18.0 m/min                xx  32    38    065    29    Y    x
x   Status:    3D FIX (28 secs)          xx 120    33    211    00    N    x
x   Longitude Err: +/- 25 m             xx  27    18    157    30    Y    x
x   Latitude Err: +/- 24 m             xx  10    12    055    41    Y    x
x   Altitude Err: +/- 20 m            xx  11    00    000    32    N    x
x   Course Err: n/a                    xx                                     x
x   Speed Err: +/- 181 kph             xx                                     x
x   Time offset: 0.355                 xx                                     x
x   Grid Square: JN35vc                xx                                     x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqjmqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj

```

Figura 4.6: Dati in output al comando `cgps -s`. Nella colonna in alto a sinistra si notano le coordinate nello spazio e altri dati utili alla navigazione. Da notare, inoltre, come il tempo indicato nella prima riga sia ora Zulu, che è l'ora di Greenwich (UTC). In Italia, l'ora corrisponde a UTC+1 (ora solare, autunno/inverno) oppure UTC+2 (ora legale, primavera/estate). Quindi il test è stato fatto alle 16:34.

acquisizione dei dati, filtraggio e calibrazione scritte dalla tesi precedente a questa[7]. Nell'utilizzo delle librerie sopra-descritte non è possibile visualizzare direttamente a schermo il corretto funzionamento della piattaforma, poiché i dati devono essere prima filtrati. Quindi per poter leggere i dati dei sensori in maniera semplice viene utilizzato un programma che permette di eseguire questi comandi. Questo software si è `minimu9-ahrs`.

### 4.4.1 `minimu9-ahrs`

`minimu9-ahrs` (ahrs sta per attitude and heading reference system) è un programma di lettura dei dati dai sensori di Pololu MinIMU-9 v3 e schede simili attraverso I<sup>2</sup>C. Questo software serve esclusivamente a visualizzare i dati per capire come funzionano i sensori, che sensibilità e orientamento hanno, ma non può essere utilizzato nei calcoli degli angoli di Eulero finali, che verranno estratti attraverso un filtro dedicato. L'installazione e l'utilizzo di `minimu9-ahrs` avviene per mezzo di GitHub [17]. Eseguita l'installazione è possibile calibrare la piattaforma con dei comandi dedicati, leggere i dati grezzi provenienti dalla piattaforma oppure, una volta filtrati, leggerli in termini degli angoli di Eulero. Un esempio di output di dati che scorrono sullo schermo in termini di angoli di Eulero si ha nella figura 4.7. Il comando dalla linea di comando per avere questi dati è `minimu9-ahrs -output euler`.

### 4.4.2 `ahrs-visualizer`

`ahrs-visualizer` è una implementazione grafica del `minimu9-ahrs`, che basa il suo funzionamento su un sottoinsieme di librerie grafiche. L'installazione e utilizzo del programma[4] è simile a quello di `minimu9-ahrs` e deve essere utilizzato insieme a quest'ultimo. Vi è la necessità di collegarsi a uno schermo dal Raspberry per visualizzare l'orientamento. Al comando: `minimu9 - ahrs | ahrs - visualizer`, si ha un risultato indicato nella figura 4.8. Muovendo la piattaforma in diverse direzioni,

Yaw	Pitch	Roll
1.579	2.240	4.280
1.566	2.239	4.271
1.558	2.240	4.271
1.533	2.256	4.266
1.529	2.251	4.266
1.529	2.256	4.269
1.544	2.270	4.286
1.534	2.275	4.284
1.523	2.292	4.281
1.503	2.309	4.275
1.516	2.288	4.285
1.535	2.261	4.299
1.531	2.265	4.297
1.509	2.269	4.284
1.500	2.273	4.282
1.486	2.288	4.280
1.507	2.263	4.296
1.490	2.256	4.291
1.466	2.268	4.272
1.441	2.297	4.258
1.458	2.304	4.276
1.490	2.287	4.290
1.525	2.294	4.312
1.548	2.292	4.316
1.546	2.295	4.300
1.526	2.289	4.277
1.507	2.277	4.262

Figura 4.7: Gli angoli di Eulero visualizzabili con il programma *minimu9-ahrs*. Un'imbardata positiva ( $\text{yaw} > 0$ ) corrisponde a una rotazione oraria attorno all'asse Z vista dall'alto. Un beccheggio positivo ( $\text{pitch} > 0$ ) corrisponde a una rotazione attorno all'asse Y che causerebbe all'asse X di puntare verso l'alto. Un rollio positivo ( $\text{roll} > 0$ ) corrisponderebbe a una rotazione antioraria attorno all'asse X. Dai dati si nota che la piattaforma ha un angolo di circa un grado e mezzo più a est rispetto al Nord Magnetico, punta in alto di poco più di due gradi ed è girato attorno all'asse longitudinale di poco più di quattro gradi verso sinistra rispetto a un o piano orizzontale.

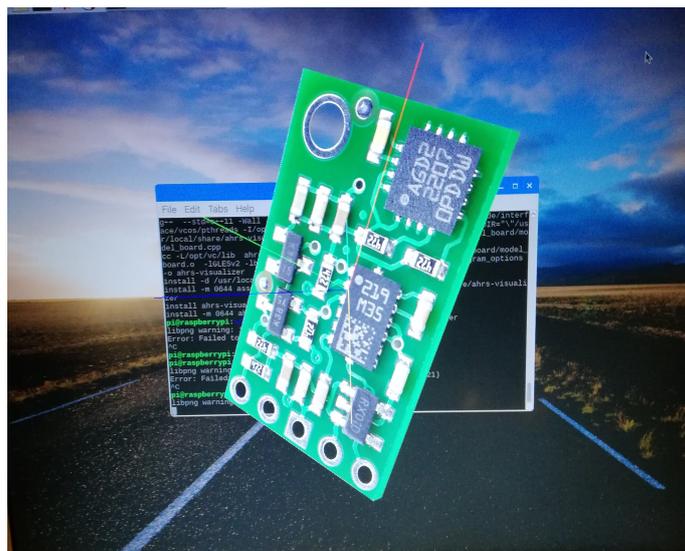


Figura 4.8: Applicazione del *ahrs-visualizer*.

l'immagine sullo schermo si muove di conseguenza. È un applicativo puramente visivo, ma che ci permette di capire meglio il funzionamento della piattaforma inerziale.

### 4.4.3 Introduzione al filtro di Madgwick

I dati provenienti dalla piattaforma inerziale, così come sono, non possono essere utilizzati nei nostri calcoli, poiché sono dati grezzi, affetti da errori e rumori, e perciò devono essere elaborati, o meglio filtrati in maniera opportuna, trovando un compromesso tra costo computazionale e precisione.

Nella letteratura vi sono presenti diversi tipi di filtri, con gradi di precisione più elevati a scapito della velocità di esecuzione, come ad esempio i filtri di Kalman<sup>6</sup> oppure con una più grande velocità di elaborazione, a scapito della precisione dei risultati, come il qui introdotto filtro di Madgwick.

Questo filtro è stato qui utilizzato in modo pratico, senza approfondire il suo funzionamento nei dettagli [24].

Il filtro di Madgwick è un algoritmo open-source [14] che calcola, dal punto di vista computazionale, in maniera efficiente gli angoli di assetto, a partire da unità di misura inerziali (IMU) costituite da sensori di tipo MARG (Magnetic, Angular Rate, and Gravity) consistenti di giroscopi, accelerometri e magnetometri tri-assiali. Questi sensori includono anche una compensazione della distorsione magnetica. L'algoritmo si basa sull'utilizzo dei quaternioni<sup>7</sup> permettendo ai dati provenienti dall'accelerometro e magnetometro di essere utilizzati in un algoritmo che calcola la misura dell'errore del giroscopio sotto forma di derivata del quaternione. Sono state fatte sia valutazioni empiriche sulle prestazioni di tale algoritmo, confrontandole con sensori di orientamento commerciali, sia valutazioni messe a confronto con algoritmi di orientamento basati su filtro di Kalman. I risultati indicano che l'algoritmo di Madgwick raggiunge livelli di accuratezza in confronto con quello di Kalman relativamente buoni: un errore quadratico medio  $< 0.8^\circ$  nel caso statico e  $< 1.7^\circ$  nel caso dinamico. Il basso carico computazionale, unito alla capacità di svolgere i calcoli a una bassa velocità di campionamento, evita l'utilizzo di grandi equipaggiamenti ed energie, consentendo la creazione di sistemi leggeri, poco costosi e duraturi [24, p. 5].

L'algoritmo di Madgwick da in output i valori dei quattro quaternioni. Con delle relazioni trigonometriche possono essere calcolati gli angoli di Eulero.

## 4.5 Maestro

La preparazione all'utilizzo del Maestro avviene in due passaggi:

- Configurazione del Maestro e inizializzazione dei servi
- Connessione del Maestro con il Raspberry Pi e le funzioni di dialogo

---

<sup>6</sup>Il filtro di Kalman è un filtro ricorsivo che stima lo stato di un sistema dinamico a partire da una serie di misure soggette a rumore. È un filtro efficiente, ma ha costi computazionali relativamente elevati.

<sup>7</sup>I quaternioni sono una estensione dei numeri complessi. Poiché possono rappresentare uno spazio vettoriale reale di dimensione 4 (analogamente ai complessi, che sono uno spazio a 2 dimensioni, cioè un piano) di tipo  $a+bi+cj+ck$ , trovano molte applicazioni nella modellizzazione delle rotazioni dello spazio.

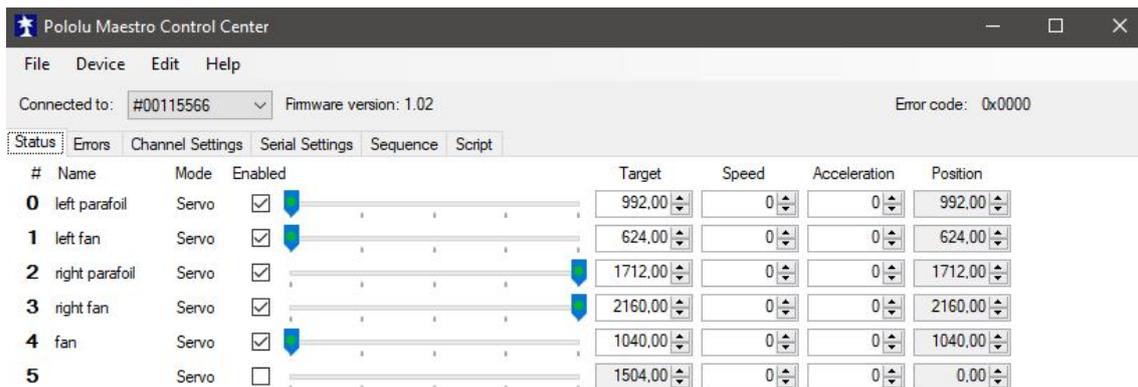


Figura 4.9: Schermata del Pololu Maestro Control Center per Windows, installato sul proprio computer. Serve per configurare e testare, graficamente, il movimento dei servomeccanismi.

## 4.5.1 Configurazione del Maestro e inizializzazione dei servi

Il primo passo importante per la configurazione del dispositivo della Pololu Micro Maestro 6-Channel USB Servo Controller [21], consultando il Manuale dell'utilizzatore [16], è scaricare dal sito ufficiale il **Pololu Maestro Control Center**, nel nostro caso, per Windows [15] e installarlo sul proprio PC. Esso è uno strumento grafico (figura 4.9) che facilita l'uso del Maestro attraverso l'interfaccia USB. Per quasi tutti i progetti è necessario iniziare a usare il centro di controllo per configurare e testare il movimento dei servi collegati.

Si ricorda che il collegamento USB alimenta solamente l'elettronica di bordo del Maestro e non alimenta i servomeccanismi. Per fare ciò bisogna collegare un'alimentazione in parallelo sui pin dedicati. Nel nostro caso utilizzeremo il BEC dell'ESC che, oltre a trasmettere i dati dello stato del controllore di velocità, alimenta in parallelo tutti gli altri servomeccanismi.

### 4.5.1.1 Status (Maestro)

Nella sezione **Status** si nota nella prima colonna il numero identificativo per ogni canale del Maestro, corrispondente a un servomeccanismo. Si possono cambiare i nomi alla propria preferenza. Abbiamo chiamato i canali come segue:

- Canale 0: *left parafoil* corrisponde al servomeccanismo di sinistra che andrà a muovere le linee di controllo del Parafoil
- Canale 1: *left fan* corrisponde al servomeccanismo di sinistra che andrà a inclinare la ventola intubata sinistra
- Canale 2: *right parafoil* corrisponde al servomeccanismo di destra che andrà a muovere le linee di controllo del Parafoil
- Canale 3: *right fan* corrisponde al servomeccanismo di destra che andrà a muovere inclinare la ventola intubata destra
- Canale 4: *fan* corrisponde al collegamento con i motori delle ventole intubate. Da questo canale arriva anche l'alimentazione di tutti i servi

Se si vuole attivare un certo canale spuntare l'abilitazione nella colonna denominata **Enabled**. Ad abilitazione attivata si può muovere il cursore sulla riga di controllo. Nel nostro caso, abbiamo attivati 5 canali su 6 totali (da 0 a 4). Se un canale è configurato come **servo** (per default) l'attivazione della colonna **Enabled** permette di avere un output, quindi il movimento del servo. Se si trascina il cursore si regola il target del canale, mentre la pallina verde indica la posizione corrente del canale. Per un controllo più preciso, si può inserire un valore nominale direttamente nella casella "**Target**". La lunghezza della barra di scorrimento viene ridimensionata automaticamente con i limiti i valori minimi e massimi specificati nella sezione "**Channel Settings**".

#### 4.5.1.2 Channel Settings (Maestro)

La sezione **Channel Settings**(nella figura 4.10) ci da un serie di informazioni importanti, di cui dobbiamo tenere conto, poiché i parametri qui impostati saranno attivati all'avvio del Maestro. In particolare:

1. **Name**: serve per cambiare i nomi dei canali dei servi; il nome, tuttavia, rimane salvato non sul Maestro, ma solamente sul computer che viene utilizzato.
2. **Mode**: è il parametro di base. Vi sono 3 opzioni: **Input**, **Output** e **Servo**. Noi utilizzeremo solamente il terzo. **Servo** indica l'output di un servo di tipo R/C (Radio Control), controllato tramite un segnale di tipo PWM (*pulse-width modulation*): infatti, nei moderni servo RC l'angolo di rotazione meccanico è determinato dalla larghezza di questo impulso elettrico che viene applicato al filo di controllo. La larghezza dell'impulso determinerà fino a che punto potrà girare il motore. Questo impulso viene emanato ogni 20 ms (50 Hz) e il loro valore viene definito in  $\mu s$ .
3. **Min** e **Max** (espressi in  $\mu s$ ): l'escursione dei bracci dei servi va da un minimo di  $0^\circ$  a un massimo di  $180^\circ$  circa. Il valore di minimo corrisponde a circa 1 ms (1000  $\mu s$ ), il massimo a circa 2 ms (2000  $\mu s$ ). Si impostano i valori in maniera empirica, in funzione del movimento che devono fare i servi, ma i valori di minimo e massimo si aggireranno intorno a questi valori. Nella figura 4.11 viene indicato uno schema di come funziona la rotazione del braccio del servo alla variazione della larghezza dell'impulso.
4. **On start-up or error**: specifica a quale valore si deve portare il PWM quindi il servo in caso di accensione o errore. Noi utilizziamo l'opzione **Go to**, che permette di specificare una posizione di default per ogni servo. In questo modo diamo dei valori iniziali della posizione dei servi, ma soprattutto dell'ESC, che viene inizializzato a un numero di  $\mu s$  tale da evitare l'accensione delle ventole in maniera inaspettata.

Una volta impostati questi valori, essi vengono salvati a parte, perché serviranno durante l'esecuzione del programma principale con l'ausilio di due funzioni, scritte in C, che sia leggono i valori dei segnali sia li comandano, direttamente dentro il programma principale.

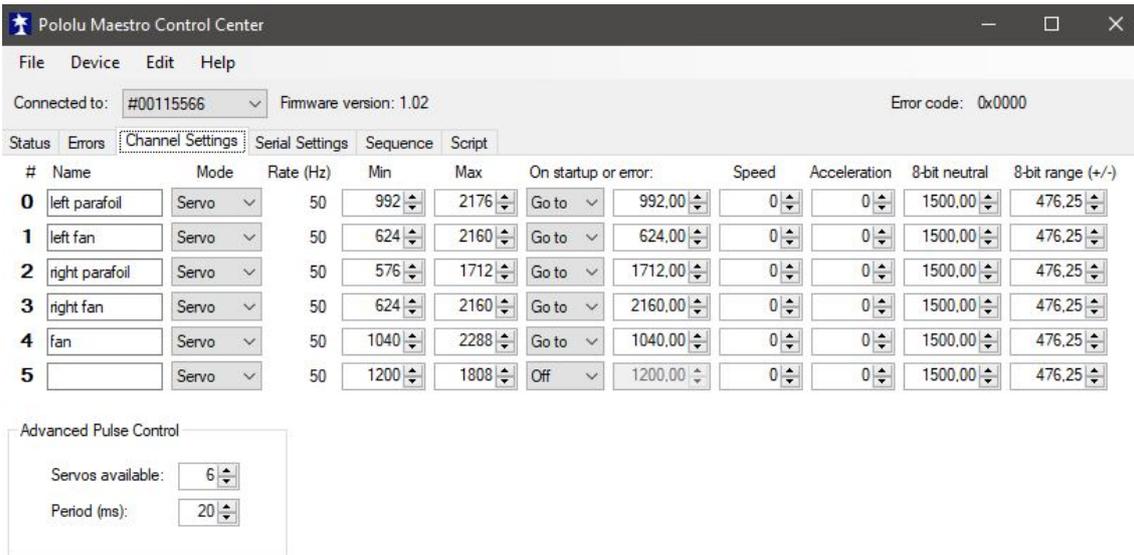


Figura 4.10: L'impostazione dei canali serve per inizializzare i servi all'avvio.

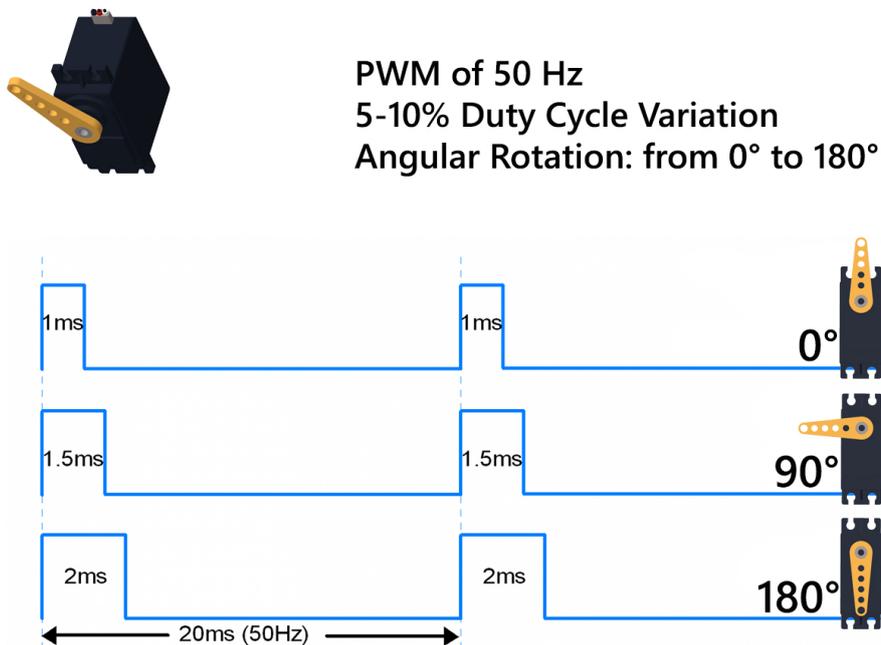


Figura 4.11: Variazione dell'angolo di rotazione del servo al variare del segnale PWM (Pulse Width Modulation) espresso in  $\mu\text{s}$ . Essi si aggirano attorno a  $1000 \mu\text{s}$  in posizione minima e  $2000 \mu\text{s}$  in posizione massima, con tutti i valori intermedi. La frequenza con cui vengono emessi gli impulsi sono di 50 Hz. Per ogni servo si impostano i valori di minimo e massimo.

## 4.5.2 Connessione del Maestro con il Raspberry Pi

La connessione a livello informatico si ha attraverso un algoritmo scritto in C [25], sfruttando gli standard POSIX (Portable Operating System Interface for Unix), in particolare utilizzando le librerie `<termios.h>`, `<fcntl.h>` e `<time.h>`, che permettono rispettivamente di interfacciarsi con terminali di dati seriali, di leggere e scrivere file e vedere le funzioni riguardanti tempo e date.

Le funzioni sono due:

- **maestroGetPosition**: restituisce il valore in  $\mu s$  della posizione di un certo canale dato in input; in input viene dato anche un flag di corretta apertura del dispositivo. Input: il numero del canale e il flag; Output: il numero di microsecondi che corrisponde a una certa posizione.
- **maestroSetTarget**: legge il valore di un certo target in  $\mu s$  per un certo canale, dati entrambi in input, e lo porta a tale valore; in input viene dato anche un flag di corretta apertura del dispositivo. Input: il valore in microsecondi che vogliamo raggiungere, il canale che deve essere attivato e il flag. Output: niente, solo il movimento del servo.

I valori che vengono dati in pasto a queste funzioni sono quelli calcolati empiricamente nella sezione precedente. Così, per ogni canale, abbiamo dei numeri precisi per le posizioni che vogliamo raggiungere. È necessario pertanto salvare a parte questi valori per ogni configurazione. Qualora cambiassero i servi o il loro collocamento sulla sonda, lo spostamento dei bracci non sarà lo stesso.

## 4.6 Software per il controllo del sistema

Ora che tutte le interfacce sono connesse e funzionanti, dobbiamo pensare a un metodo di comando del sistema di rientro, che ci permetta di muovere i suoi servi in modo da cambiare la sua traiettoria.

Sono stati creati due programmi, in C, in grado di cambiare la traiettoria della sonda. Essi sono:

1. Controllo autonomo della direzione di volo
2. Controllo manuale della traiettoria

### 4.6.1 Controllo autonomo della direzione del volo

È un programma già sviluppato nei lavori precedenti [7]. Il funzionamento consiste nel chiedere all'utente di immettere da tastiera un angolo di navigazione rispetto a un certo riferimento, nel nostro caso il Nord indicato dalla piattaforma inerziale. Chiamiamo questo angolo  $yaw_{ref}$ , ed è fisso una volta inserito. Il programma, a quel punto, inizia a muovere i servi che comandano le linee di controllo del parafoil finché non viene annullata la differenza tra  $yaw_{ref}$  e l'angolo che si ha rispetto al Nord e la direzione di avanzamento della sonda. Questo angolo viene chiamato  $yaw$ , ed è ricavato ad ogni ciclo come output dalla piattaforma inerziale, filtrato attraverso l'algoritmo di *Madgwick*. La differenza tra i due angoli (che è anche l'errore:  $err = |yaw - yaw_{ref}|$ ) non sarà mai nulla, perché vi è sempre presente

un'oscillazione dei valori, legata alla precisione del dispositivo e alle approssimazioni dei calcoli numerici e ai rumori di varia natura. Questo errore, ad ogni ciclo di calcolo, moltiplicato per un certo guadagno, viene utilizzato per variare l'intensità con cui viene cambiata la posizione del servo. Ad esempio se l'errore è piccolo, quindi siamo vicini alla direzione di volo desiderata, i servi si muoveranno di poco, se invece l'errore è grande i servi avranno un movimento accentuato per cercare di annullare il prima possibile l'errore. Se l'errore rimane entro un certo range di valori tollerabili, potremmo considerare il risultato soddisfacente. Nonostante questo software sia limitato all'annullamento dell'errore tra due angoli di navigazione, è un primo approccio per lo sviluppo di un software che permetta un pilotaggio autonomo della sonda.

## 4.6.2 Controllo manuale della traiettoria

Questo software è stato scritto *ex novo* e permette il pilotaggio della sonda di rientro da tastiera del computer. Il programma, una volta avviata la connessione in remoto con l'unità GNSS, IMU e il Maestro, a cui sono connessi i servi e il controllo dei motori, richiede che siano premuti certi tasti in particolare. I tasti sono stati scelti in modo da facilitare il pilotaggio da parte dell'utente. Essi possono essere premuti sia in minuscolo che maiuscolo e sono:

- **Y** per azzerare la spinta delle ventole
- **U** per spingere al 25% della spinta totale
- **I** per spingere al 50% della spinta totale
- **O** per spingere al 75% della spinta totale
- **P** per spingere al 100% della spinta totale
- **W** per muovere entrambi i servi del Parafoil in sù (4.12 (a))
- **A** per muovere il servo sinistro del Parafoil in giù e quello destro in sù (la sonda vira a sinistra) (4.12 (b))
- **Q** per muovere il servo sinistro del Parafoil in giù e quello destro in sù. In più la ventola sinistra punta in avanti, mentre quella di destra punta indietro. (la sonda vira in modo vigoroso a sinistra)
- **S** per muovere entrambi i servi del Parafoil in una posizione intermedia tra massimo sù e massimo giù
- **D** per muovere il servo destro del Parafoil in giù e quello sinistro in sù (la sonda vira a destra)(4.12 (c))
- **E** per muovere il servo destro del Parafoil in giù e quello sinistro in sù. In più la ventola destra punta in avanti, mentre quella di sinistra punta indietro. (la sonda vira in modo vigoroso a destra) (4.12 (d))
- **X** per muovere entrambi i servi del Parafoil in giù
- **J** per girare le ventole in avanti (permette al Parafoil di creare la portanza)



(a)



(b)



(c)



(d)

Figura 4.12: Movimentazione dei servi del parafoil. (a) Posizione neutra, si riposiziona in questa configurazione tutte le volte che viene acceso il sistema. Entrambe le leve del parafoil volte in su (tasto **W** da tastiera), i fan rivolti in avanti. (b) La leva del servo sinistro girato in giù, quello di destra in su (tasto **A** da tastiera), i fan sono sempre rivolti in avanti. Il sistema vira a sinistra. (c) La leva del servo destro girato in giù, quello di sinistra in su (tasto **D** da tastiera), i fan sono sempre rivolti in avanti. Il sistema vira a destra. (d) Una possibile combinazione: la leva destra del parafoil in giù, la leva sinistra del parafoil in su, motore destro rivolto in avanti, motore sinistro rivolto indietro. Provoca una virata stretta a destra. Tasto **E** da tastiera in questo caso. Per riportare le ventole in posizione avanti (neutra per noi) utilizzare il tasto **J** da tastiera e il tasto **W** per le leve del parafoil.

- **H** per girare le ventole di circa 45 gradi in sù (permette al Parafoil di creare meno portanza, ma alleggerisce il carico che regge) (4.13 (b))
- **K** per girare le ventole in sù di 90 gradi rispetto all'avanti (sgrava momentaneamente il peso) (4.13 (a))
- **L** per girare le ventole indietro di circa 180 gradi rispetto all'avanti (frena bruscamente) (4.13 (c))
- **T** per terminare le operazioni e uscire dal programma

## 4.7 Implementazione complessiva dei software

Nella figura 4.14 viene esposto schematicamente il collegamento logico tra i vari programmi utilizzati a proposito del pilotaggio del sistema di rientro.

La sonda, a collegamento con i dispositivi riuscito, deve gestire una serie di informazioni. Si è creato un programma unico, in cui i programmi singoli di ogni dispositivo, per quanto possibile, convergono. Lo scopo è quello di conoscere a ogni istante la propria posizione e il proprio assetto grazie ai dati forniti dall'unità GPS e quella della piattaforma inerziale e di poter utilizzare i servi nella maniera più opportuna, come ad esempio tenere una certa traiettoria in maniera autonoma oppure pilotarlo a vista. I segnali informatici provenienti dai dispositivi sono gestiti grazie a una serie di librerie dedicate, che vanno incluse nel *header* del programma principale come delle librerie normali. Vengono utilizzate anche alcune librerie classiche, come ad esempio `math.h` e `string.h`, per rispettivamente i calcoli matematici e la gestione di eventuali testi. Le funzioni delle librerie dedicate hanno un ruolo importante nel corretto funzionamento del flusso di dati e sono state spiegate nelle sezioni precedenti.

Dalla figura 4.14 si possono notare quattro porte di connessione con il programma principale, denominato `main`. Tutte arrivano a comunicare con la parte del controllo remoto, che decide le azioni da intraprendere. In particolare abbiamo:

- **GPS.** Attraverso la connessione di tipo *UART* arriva il segnale al Raspberry. Grazie alla libreria `wiringPi.h` il segnale dei dati diventa informazione, ma in un formato di tipo *NMEA-0183*, il quale viene trasformato per mezzo della libreria `gps.h` in un formato a noi noto, cioè in termini di **latitudine**, **longitudine** e **altitudine**. Si hanno queste informazioni solo se si ha una connessione stabile con almeno quattro satelliti. Nel `main` prima di procedere si fa il controllo sulla stabilità della connessione.
- **IMU.** La libreria `wiringPi2C.h` contiene una serie di funzioni che possono leggere i registri del bus *I<sup>2</sup>C* e quindi trasformare i flussi di bit della piattaforma inerziale in dati utili. Si ha così a disposizione un set di tre dati per ogni asse. Le accelerazioni  $a_x$ ,  $a_y$ ,  $a_z$ , le velocità giroscopiche  $g_x$ ,  $g_y$ ,  $g_z$  e intensità dei campi magnetici  $m_x$ ,  $m_y$ ,  $m_z$ . Questi dati, in realtà, devono essere corretti di certi valori di offset, poiché soffrono di disturbi sistematici (*bias*). Per fare ciò si eseguono delle calibrazioni per il giroscopio e per il magnetometro. A questo punto entra in gioco il filtro di Madgwick, richiamato nel `main` da `#include "MadgwickAHRS.h"`. Esso prende in input le grandezze di sopra e da in output i quattro quaternioni. Con delle relazioni trigonometriche note e



(a)



(b)



(c)

*Figura 4.13: Movimentazione delle ventole intubate. (a) Entrambe le ventole volte in su (tasto **K** da tastiera). (b) Entrambe le ventole girate di circa  $45^\circ$  in su (tasto **H** da tastiera). (c) Entrambe le ventole volte all'indietro, utili in caso di frenata brusca in direzione orizzontale (tasto **L** da tastiera). Per riportare le ventole in posizione avanti (neutra per noi) utilizzare il tasto **J** da tastiera*

con una funzione di conversione da radianti a gradi si ottengono i tre angoli di assetto: *yaw*, *pitch* e *roll*.

- **Maestro.** Le librerie `termios.h` e `fcntl.h`, accompagnate dalle funzioni `maestroGetPosition` e `maestroSetTarget` permettono al Maestro di comunicare con il `main`. I valori iniziali per la posizione dei servi sono stati calcolati esternamente ed empiricamente.
- **User.** Con un approccio classico di input da tastiera l'utente può decidere di utilizzare il controllo in due modi. Il primo è il pilotaggio autonomo, dando in input solamente l'angolo di navigazione rispetto al Nord a cui la sonda, attraverso la movimentazione dei bracci del Parafoil, si deve allineare. Il secondo è il pilotaggio manuale della sonda attraverso l'utilizzo della tastiera, come descritto nella sezione precedente. Nei nostri test preliminari utilizzeremo questa seconda modalità.

Ad ogni ciclo di calcolo, i dati di posizione, assetto e posizionamento dei servi vengono aggiornati e, se è necessario, viene corretta la traiettoria con dei comandi automatici al Maestro. Questa opzione è disponibile solamente solo se si è nella condizione di volo autonomo: con un controllo di tipo `if` si cerca di ridurre entro un certo range l'angolo di imbardata di riferimento e quello attuale del sistema di rientro. I dati dai sensori, salvati in un file nel formato `.csv`, verranno in seguito utilizzati nel `post-processing`.

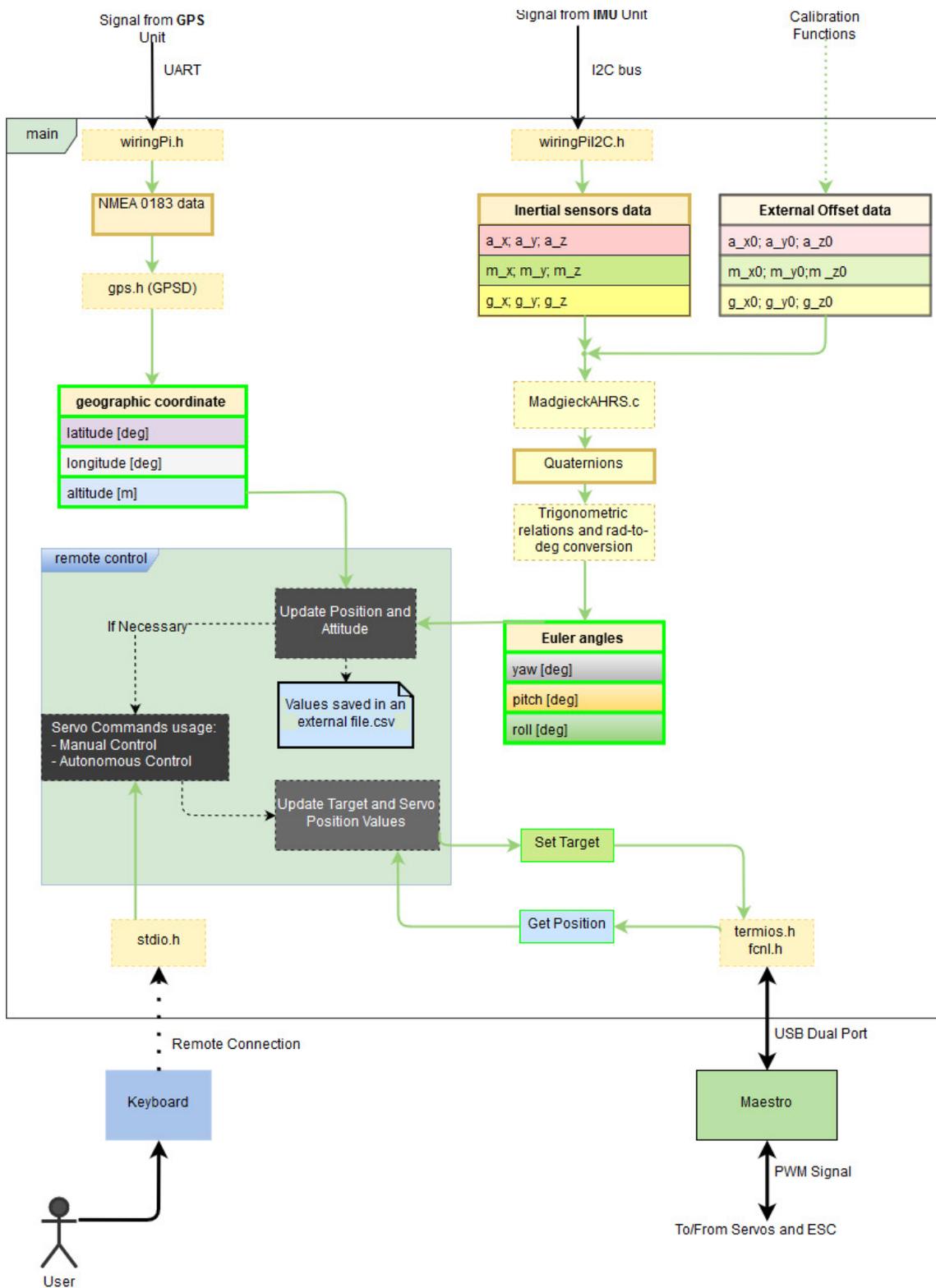


Figura 4.14: Funzionamento logico del software di pilotaggio del sistema di rientro.



## 5 | Test e Convalida

In questo capitolo sarà descritto lo svolgimento di tre test preliminari: il primo riguarda la durata della batteria, che va a confrontarsi con i calcoli teorici fatti nel capitolo sull'integrazione; il secondo riguarda la compatibilità elettromagnetica della piattaforma inerziale; il terzo riguarda il corretto funzionamento dell'unità GPS. Infine un ultimo test riguarderà la capacità di controllo della sonda da parte dell'utente. Per questo quarto test verranno esposte le calibrazioni iniziali necessarie e le eventuali correzioni svolte per migliorare le prestazioni. Infine si cercherà di rappresentare graficamente i valori assunti dalle accelerazioni, velocità giroscopiche e angoli di assetto a bordo del sistema.

### 5.1 Test sulla durata della batteria

Nella sottosezione 3.1.3 sono stati svolti i calcoli teorici sulla batteria. A pieno regime, con le entrambe ventole funzionanti in maniera continuativa, la durata della batteria era di **quasi sei minuti**.

Per la conferma di tali numeri, si è svolto un test che affaticava al massimo l'intero sistema. Le ventole erano impostate alla massima spinta, i sensori erano attivi e i quattro servomeccanismi erano mossi in maniera casuale in ogni direzione possibile e quasi in maniera continuativa.

Avviato il sistema, i dati venivano presi ogni trenta secondi, grazie a un dispositivo di tipo *tester* chiamato *cicalino* per allarme della batteria, mostrato nella figura 5.1, che indicava sul display il voltaggio della batteria sia per ogni cella che quello complessivo. Sotto certi voltaggi, nel nostro caso 3.3 V, esso comincia ad emettere un suono acuto, indicante lo stato scarico della batteria.

Durante il test, purtroppo, la ventola del motore sinistro si è staccata dall'albero del motore intorno ai 30 secondi dall'inizio del test. Si è deciso di continuare con il test. Prelevati i dati, si è interpolato linearmente e si è avuto il grafico nella figura 5.2. La durata della batteria è risultata di circa undici minuti. Quasi il doppio di quanto calcolato inizialmente. Ma, ricordando che il test è stato svolto, quasi interamente, con un solo motore, il risultato è concorde con i calcoli teorici. Infatti, un motore assorbe circa il 47% della corrente totale assorbita, quasi la metà. Se avessimo utilizzato entrambi i motori, la durata della batteria si sarebbe dimezzata, andando verso i cinque minuti e mezzo, come teorizzato. I dati sono nella tabella 5.1.

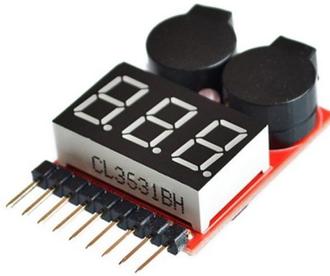


Figura 5.1: Cicalino. È un tester del voltaggio di un dispositivo, in generale della batteria. Emette un suono acuto, quando il voltaggio scende sotto una certa soglia, dando così l'indicazione che la batteria è scarica e che quindi, per non danneggiarla, conviene fermare l'utilizzo.

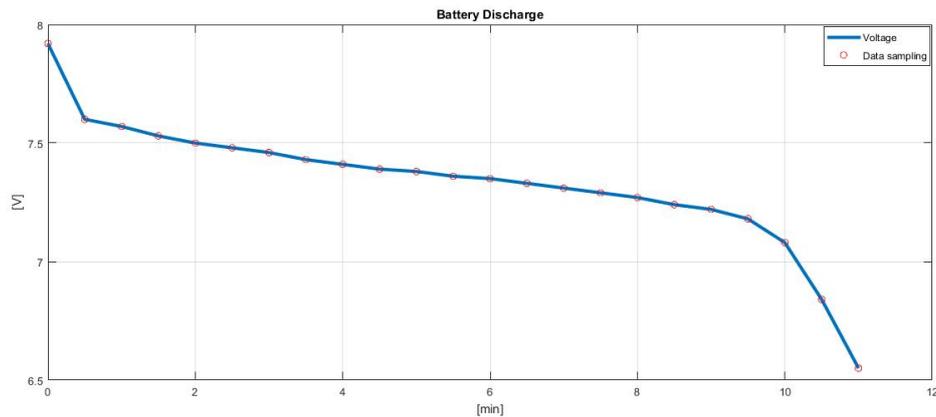


Figura 5.2: Andamento del voltaggio della batteria. I punti rossi sono i dati reali prelevati con il cicalino durante il test ogni 30 secondi. La curva blu è l'interpolazione lineare sui dati reali. Si ricorda che il test è stato fatto con una sola ventola accesa. Il cicalino ha cominciato a suonare attorno a 11 minuti dall'inizio del test. Un motore assorbe circa il 47% della corrente totale, quasi la metà. Se avessimo avuto entrambi i motori accesi, la durata della batteria sarebbe stata quasi dimezzata, confermando i calcoli teorici.

Tabella 5.1: Durata della batteria in caso teorico e in caso di test. Nel primo caso è stato calcolato per entrambi i motori accesi, mentre nel caso pratico si è utilizzato solo un motore.

Teoria (2 motori) [min]	Test (1 motore) [min]
5.77	11.0

## 5.2 Test sulla compatibilità elettromagnetica dell'IMU

L'attenzione dedicata alla compatibilità elettromagnetica è dovuta al fatto che la piattaforma inerziale contiene un magnetometro utile alla determinazione del Nord magnetico terrestre e quindi all'assetto dal punto di vista dell'imbardata (yaw). Il magnetometro viene influenzato da tutti i campi magnetici presenti vicino a esso. Per evitare che essi disturbino l'indicazione dell'angolo di imbardata bisognerà fare una serie di calibrazioni.

Vi sono stati eseguiti una serie di test su come la presenza della ventola intubata, che contiene un motore, dentro il quale si ha un magnete permanente, influisca sull'angolo di imbardata indicato dal magnetometro della piattaforma inerziale. Uno dei test consiste nel tenere ferma l'IMU e avvicinare a intervalli regolari la ventola tenendola vicino cinque centimetri per cinque secondi. Nella figura 5.3 si nota tale effetto sull'angolo di imbardata e beccheggio.

Un altro test, con effetti ben visibili, considera come variano gli angoli in funzione della distanza del magnete dalla piattaforma inerziale. Nella figura 5.4 si notano tali effetti. A circa quindici centimetri gli effetti sono minimi.

Per questo motivo prima di utilizzare i sensori è necessario fare una calibrazione in modo da annullare l'influenza dei campi magnetici provenienti dal hardware del sistema, soprattutto dai motori delle ventole intubate. Per essere ancora più precisi si dovrebbe fare altri test in cui si visualizza la differenza dell'influenza dei campi magnetici in condizione di motori spenti e accesi.

## 5.3 Test sul corretto funzionamento dell'unità GPS

Si è svolto un semplice test di tracciamento della posizione da parte dell'unità GPS a nostra disposizione. Il confronto avveniva con l'unità GPS integrata in un cellulare. Si considerano i dati provenienti da quest'ultima affidabili.

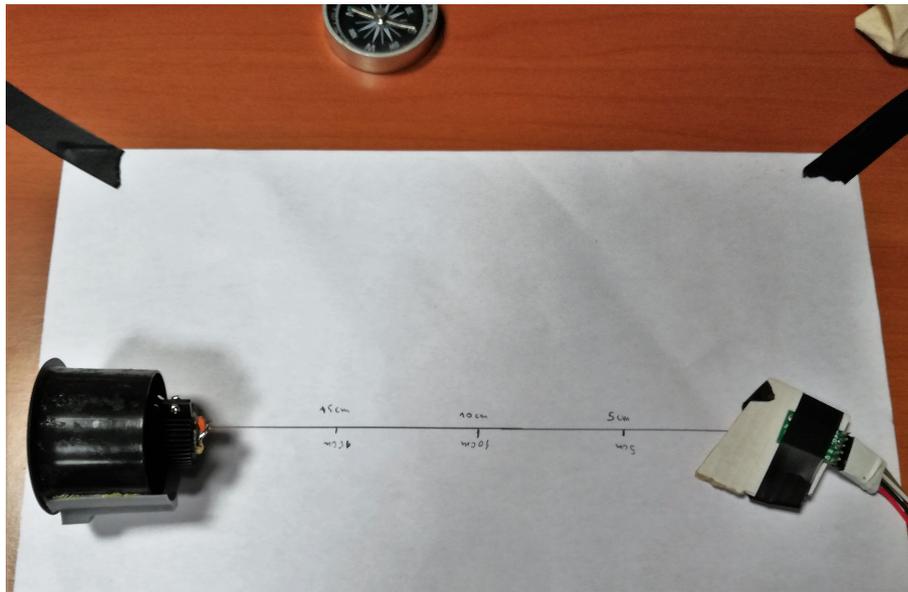
Instaurata una connessione stabile dell'unità GPS, si avvia una registrazione di dati su un file .csv. Eseguito il percorso, si confrontano i dati salvati dalla nostra unità GPS e quella presente sullo smartphone, i cui dati sono stati salvati da una applicazione apposita. Il confronto viene fatto utilizzando Google Earth Pro. I risultati si notano nella figura 5.5 e possono essere considerati soddisfacenti.

## 5.4 Preparazione al test preliminare di volo

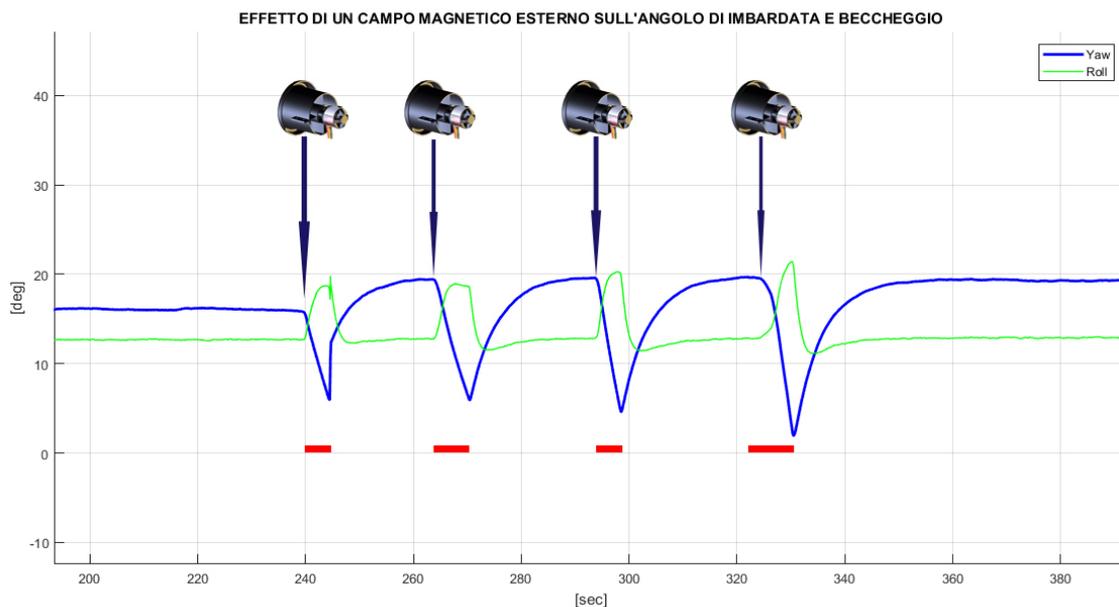
L'attivazione del hardware viene suggerita nella sottosezione 5.4.1. In seguito viene mostrato come eseguire le calibrazioni e, infine, si fanno i test preliminari sul corretto funzionamento dell'intero sistema di rientro.

### 5.4.1 Utilizzo della sonda di rientro

Come utilizzare la sonda di rientro viene indicato nella seguente lista.



(a)



(b)

Figura 5.3: Effetto di un campo magnetico esterno sugli angoli di assetto che dipendono dal magnetometro: (a) La piattaforma inerziale viene fissata a un supporto rigido, come il tavolo e viene posizionata, in modo che punti circa al Nord, utilizzando una bussola; (b) Variazione degli angoli di imbardata e rollio quando il magnete presente dentro il motore viene avvicinato a cinque centimetri ogni cinque secondi circa. I tratti in rosso indicano per quanto tempo il magnete è stato tenuto vicino alla piattaforma. La curva blu è l'andamento dell'angolo di imbardata e quella verde dell'angolo di rollio.

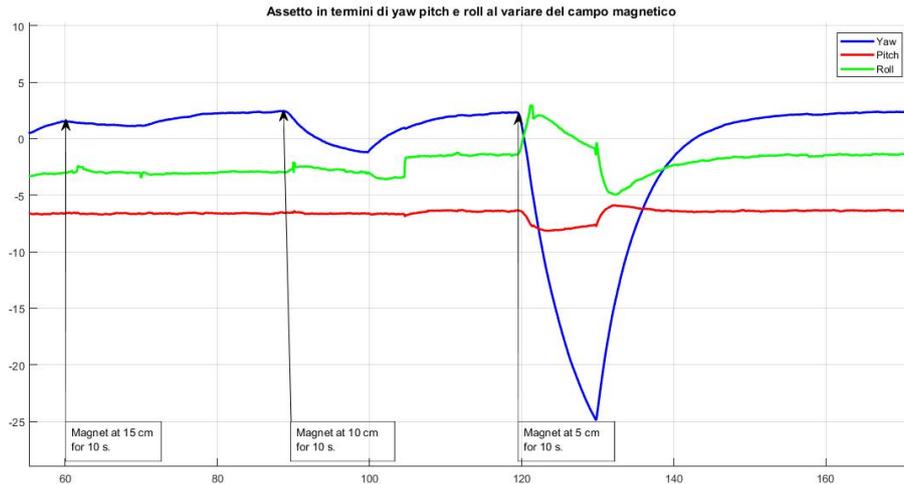
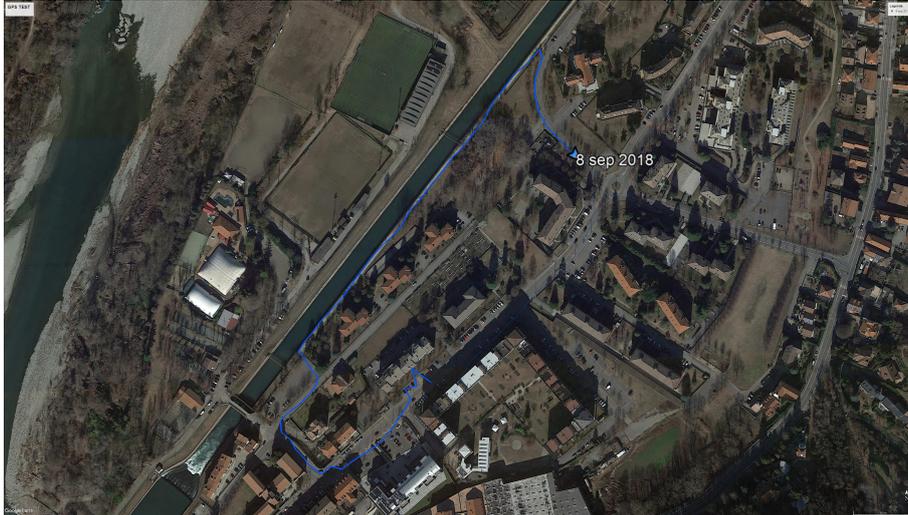


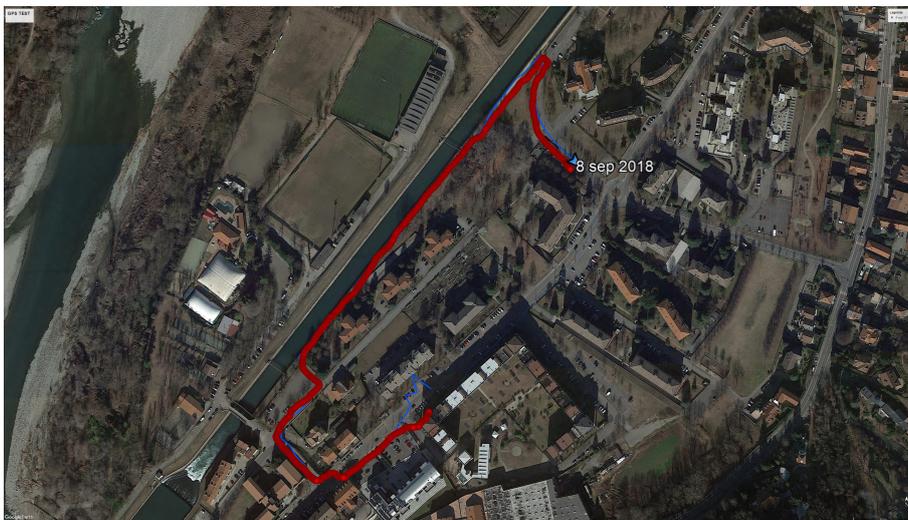
Figura 5.4: Variazione degli angoli di assetto alla variazione della distanza del campo magnetico del motore. A regime, viene avvicinata la ventola alla piattaforma inerziale a distanze man mano più brevi: 15 centimetri, 10 centimetri e 5 centimetri, e viene tenuta tutte le volte per circa 10 secondi. Si nota il grande impatto che si ha con l'avvicinamento alla piattaforma.

1. Assicurarsi che la batteria sia carica. Viene utilizzato un indicatore di carica della batteria, noto come cicalino di avvertimento. Esso si collega alla stessa presa utilizzata per la ricarica della batteria
2. Assicurarsi che il Raspberry Pi abbia la stessa connessione Wi-Fi del computer che viene utilizzato
3. Accendere l'interruttore. Lampeggiano il LED dell'unità GPS per il fix e i vari LED del Maestro. I servi e i motori si portano in una posizione iniziale, definita attraverso il centro di controllo del Maestro precedentemente
4. Aspettare che il suono proveniente dall'ESC in accensione (bip-bip-bip) si sia collegato al Maestro (un solo bip lungo), quindi niente più bip
5. Aprire il terminale PuTTY e collegarsi al dispositivo con l'indirizzo IP del RPi. Si apre un emulatore del terminale del Raspberry
6. Eseguire le calibrazioni necessarie. Ricordarsi di ricompilare i file sorgenti, affinché le modifiche abbiano luogo effettivamente. Si veda la sezione sottostante per i dettagli delle calibrazioni e della compilazione.
7. Finite le calibrazioni e le nuove compilazioni, aprire lo script bash nel formato .sh che permette di collegarsi alla porta del GPS e seguire le istruzioni.
8. Comandare i servi come indicato nella legenda [4.6.2](#).

Prima di eseguire i test bisogna calibrare la piattaforma inerziale installata sulla sonda per evitare che ci dia delle informazioni sbagliati sugli angoli di assetto. Sono state scritte delle funzioni in C che permettono di acquisire i dati dalla piattaforma ancora non calibrata. Una volta acquisiti i dati nel formato di un file .csv, vengono



(a)



(b)

*Figura 5.5: (a) Percorso tracciato secondo l'unità GPS del cellulare. (b) Percorso tracciato secondo l'unità GPS a nostra disposizione*

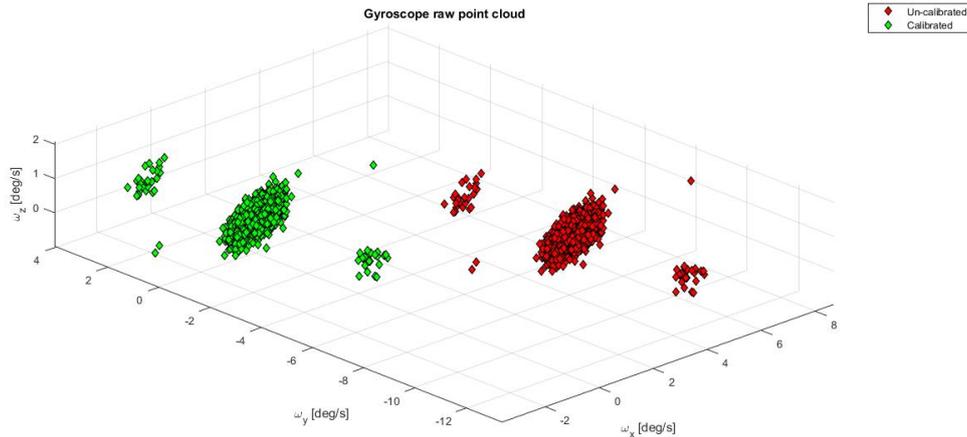


Figura 5.6: Calibrazione del giroscopio. I dati di offset vengono traslati nell'origine. I loro valori medi sono calcolati grazie a uno script Matlab. I punti rossi sono le velocità giroscopiche rilevate attorno ai tre assi. I punti verdi sono le stesse velocità, traslate nell'origine, che mediamente, quindi, hanno valore nullo.

caricati dentro dei script *Matlab* per poter estrarre i valori di offset, indicanti lo disturbi sistematici. La necessità delle calibrazioni è dovuta al fatto che alcuni dispositivi utilizzati emanano dei campi magnetici relativamente forti, distorcendo il campo magnetico che deve essere considerato dalla piattaforma inerziale. Anche le velocità giroscopiche, a dispositivo completamente fermo, presentano valori non nulli, quindi sbagliati.

## 5.4.2 Calibrazione giroscopio

Per calibrare il giroscopio, si mantiene il sistema fermo e si fa partire un programma che salva i valori delle velocità giroscopiche della piattaforma su un file `.csv`. È chiaro che le velocità, a piattaforma ferma, devono essere nulle. Quindi facendo girare per un minuto o due il programma, abbiamo le velocità medie da togliere, poiché essi presentano la *bias*, la discrepanza con i valori nominali.

Il comando da dare dal terminale da dentro la cartella `/test/imu/_test` è: `./gyroscope_test > gyroscope_test_data.csv` e lasciare ferma la sonda per un paio di minuti. Si manda via email con il programma `mpack` il file `.csv` appena creato. Utilizzando uno script Matlab si calcolano gli offset medi da impartire inizialmente per annullare gli errori sistematici. In altri termini significa traslare i dati iniziali nell'origine, in cui tutti i valori sono nulli. Un esempio di calibrazione si osserva nella figura 5.6.

## 5.4.3 Calibrazione magnetometro

Si ricorda che la calibrazione del magnetometro va fatta solo a integrazione finita del hardware, poiché ogni dispositivo elettronico presente vicino distorce il campo magnetico circostante. Vi sono due tipi di offset legati al magnetometro:

- *Hard Iron*: è legato ai campi magnetici locali, di intensità grande. Nel nostro caso la sorgente di tale effetto è dovuto ai dispositivi elettronici vicini al

magnetometro. Siamo in grado di togliere questo offset trasladando i valori del campo magnetico nell'origine.

- *Soft Iron*: è legato ai campi magnetici globali, di intensità piccola. È causato da interferenze esterne e dal campo magnetico terrestre. Non siamo in grado di annullare completamente questo offset. Infatti, dalla forma non regolare che assume il campo magnetico attorno al dispositivo, si può notare in quale direzione sia il Nord magnetico.

#### 5.4.3.1 Calibrazione iniziale da `minimu9-ahrs`

Un metodo iniziale calibrare è l'utilizzo del già installato `minimu9-ahrs`. Con il comando dal terminale `minimu9-ahrs-calibrate` si attiva la possibilità di calibrare il magnetometro: viene chiesto di muovere la piattaforma in tutte le direzioni, così che tenga conto di tutti i campi magnetici presenti vicino. Noi muoviamo l'intero sistema in tutte le direzioni finché non è detto di fermarsi. Questo è un metodo automatico e veloce per calibrare il magnetometro, ma non ci garantisce che il *bias* sia annullato. Dobbiamo agire manualmente per capire di quanto sia la discrepanza tra i valori nominali e quelli reali.

#### 5.4.3.2 Calibrazione manuale

È un metodo più lungo, ma più completo, proposto nella tesi precedente [7], che permette di capire proprio quanto i campi magnetici dei dispositivi influiscano sul corretto funzionamento del magnetometro della piattaforma inerziale.

La logica è come per la calibrazione automatica di `minimu9-ahrs`. Muovendo l'IMU in tutte le direzioni si registrano in un file `.csv` tutti i campi magnetici presenti vicino alla piattaforma. Per ogni asse, vengono presi i valori massimi e minimi e di questi valori se ne fa una media. Abbiamo così il valore medio di un campo magnetico attorno alla piattaforma. Questi valori vanno tolti dai dati iniziali per traslare il campo magnetico nell'origine.

Il comando da dare dal terminale da dentro la cartella `/test/imu/_test` è: `./magnetic_test > magnetic_test_data.csv` e muovere la sonda in tutte le direzioni, per un paio di minuti. Viene utilizzato un altro script Matlab per calcolare i valori medi per l'offset, che verranno inseriti nell'inizializzazione del file sorgente, in modo da annullare il *bias* legato al campo magnetico dei dispositivi nella vicinanza. Si annulla solo l'offset dovuto all'effetto *hard iron* trasladando il campo magnetico nell'origine (figura 5.7), quando per l'effetto *soft iron* non si può fare molto: esso ci indica la direzione del campo magnetico terrestre. Si osservi tale effetto nella figura 5.8.

#### 5.4.4 Aggiornamento Offset e Compilazione

I dati ricavati attraverso i programmi di calibrazione vengono inseriti a mano nel codice sorgente, nel *header* del file e si procede con la ricompilazione del file sorgente. La compilazione avviene da linea di comando ed è:

```
gcc -o eseguibile sorgente.c MadgiwckAHRS.c -lm -lgps -lwiringPi
```

e il loro significato è:

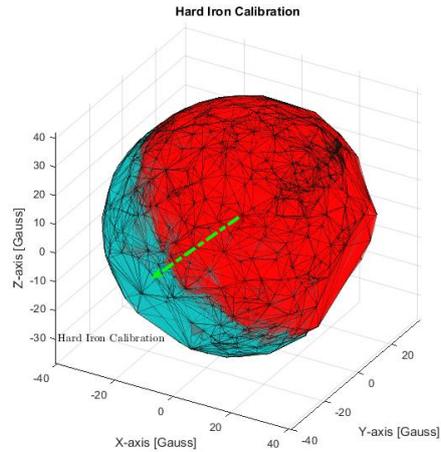


Figura 5.7: Lo slittamento del campo magnetico dovuto al hard iron. Lo sferoide rosso indica i dati grezzi provenienti dalla piattaforma inerziale. Lo sferoide azzurro è semplicemente la traslazione dello sferoide rosso nell'origine. Questa è la calibrazione di tipo hard iron.

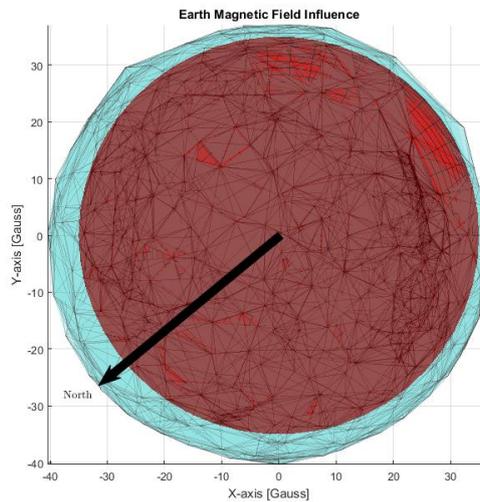


Figura 5.8: Lo slittamento del campo magnetico dovuto all'effetto chiamato soft iron. Nella direzione in cui la forma del campo magnetico è allungata, vi è il Nord Magnetico terrestre. La figura azzurra trasparente è il campo magnetico reale traslato nell'origine, quindi in assenza dell'effetto hard iron. Si nota come la sua forma sia irregolare rispetto alla sfera di colore rosso, indicante un campo magnetico uniforme attorno all'origine.

- `gcc`: GNU Compiler Collection è un compilator per i sistemi operativi GNU/Linux
- `-o`: specifica che il file oggetto che viene creato (l'eseguibile) venga messa nel file indicato subito dopo
- `eseguibile`: nome del file dentro il quale viene salvato l'eseguibile. È il file da aprire per far partire il programma
- `sorgente.c`: è il file sorgente, scritto in questo caso in C, che può essere modificato dall'utente.
- `MadgwickAHRS.c` è il file utilizzato per filtrare i dati provenienti dalla piattaforma inerziale e avere in uscita gli angoli di assetto. Comunica con il file sorgente attraverso l'inclusione nel header di questo del file `MadgwickAHRS.h`
- `-lm`: include la libreria `-math.h`, necessaria per le operazioni trigonometriche che il file sorgente deve effettuare
- `-lgps`: include la libreria `gps.h`, necessaria per estrarre le informazioni dall'unità GPS in formato a noi comodo
- `-lwiringPi`: include la libreria `wiringPi.h` e `wiringPiI2C.h`, necessarie per la trasmissione dei segnali dei dati tra le unità GPS e piattaforma inerziale e il Raspberry Pi

Eseguita la compilazione si può aprire l'eseguibile con il comando `./eseguibile` dalla linea di comando.

## 5.5 Test preliminari di volo

I test preliminari qui mostrati sono legati alla manovrabilità del sistema di rientro. Per una convalida più completa e corretta è necessario che il sistema sia lanciato da quote più alte, ad esempio da cento metri e che abbia a disposizione raggi di curvatura più ampi di quelli qui sotto descritti, per simulare in maniera più precisa il rientro del sistema dalla stratosfera.

Qui vengono indicati i test svolti solo con la seconda configurazione, le cui prestazioni vengono portati ai limiti, cercando di mantenere la quota, quando nella pratica il sistema dovrà avere una discesa controllata.

È stata eseguita una serie di test per convalidare il corretto funzionamento del sistema di controllo e dei sensori a bordo. I test sono stati svolti sia all'interno che all'esterno. La necessità di svolgerli all'esterno è legata al fatto che l'unità GNSS non riesce a comunicare con i satelliti poiché il segnale viene schermato in luoghi chiusi e per simulare un rientro un po' più realistico, con tutti i disturbi che entrano in gioco in un ambiente esterno. I test svolti all'interno mostrano come il sistema sia manovrabile in spazi relativamente ristretti.

Vengono indicati solo due test rappresentativi dei tanti svolti.



*Figura 5.9: Primo test a chiuso. Si notano, in ordine da sinistra in alto, il lancio, la stabilizzazione, una prima manovra di virata di 180° verso sinistra, una seconda virata verso destra, perdita di quota e avvicinamento all'atterraggio.*

### 5.5.1 Test di volo 1: interno

Il sistema viene lanciato da un'altezza di circa cinque metri e, dopo averlo lasciato stabilizzare per pochi secondi, si cerca di impartire dei comandi ai servomeccanismi collegati al parafoil. Essi rispondono ai comandi, tirando le linee di comando del paracadute, permettono a quest'ultimo di cambiare la sua forma, variandone la portanza. Si riesce dunque a manovrare il sistema, facendogli fare prima una virata di 180° verso sinistra, tornando indietro verso la zona di lancio, poi un comando per una virata a destra, anch'essa di 180°, per poi, infine, avvicinarsi alla parete di fronte e perdere portanza, quindi atterrare. Nella figura 5.9 vengono indicate alcune immagini in sequenza dal video del test.

### 5.5.2 Test di volo 2: esterno

Il test di volo preliminare svolto all'esterno (immagini nella figura 5.10) è servito per la convalida del corretto funzionamento dei sensori, in particolare, l'unità GPS. La meteorologia indicava: tempo sereno, vento a 6.2 km/h, umidità 36%, pressione a 989 hPa. Anche in questo caso, come nel primo test, il sistema viene lanciato da una piccola elevazione di terreno rispetto alla zona circostante, per permettergli di stabilizzarsi e acquisire velocità.

#### 5.5.2.1 Post-Processing dei dati del Test 2

Viene salvata una serie di dati in un file .csv, durante il test svolto il 26 agosto 2018. Su un'unica riga abbiamo diversi dati separati da virgola. Essi sono, nell'ordina da sinistra a destra:

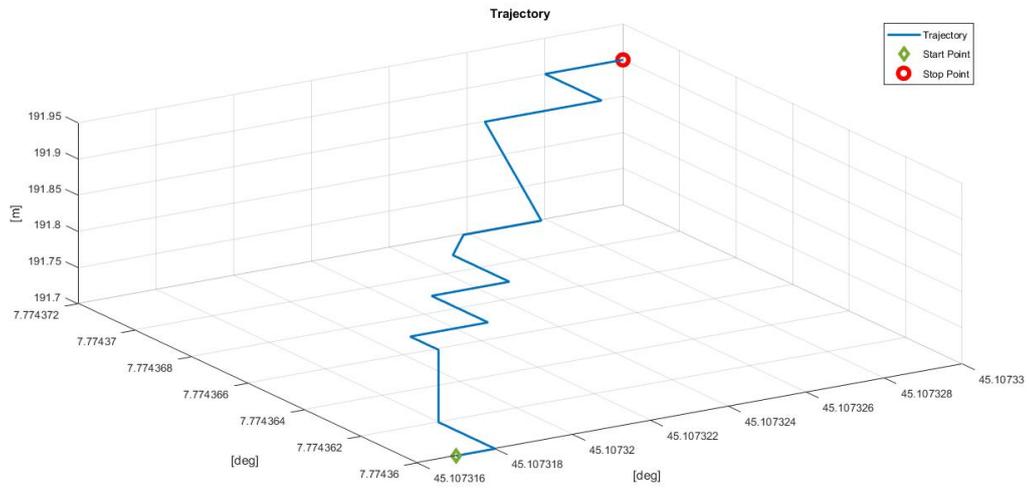
- il tempo dall'inizio dell'acquisizione dei dati
- la latitudine proveniente dall'unità GPS



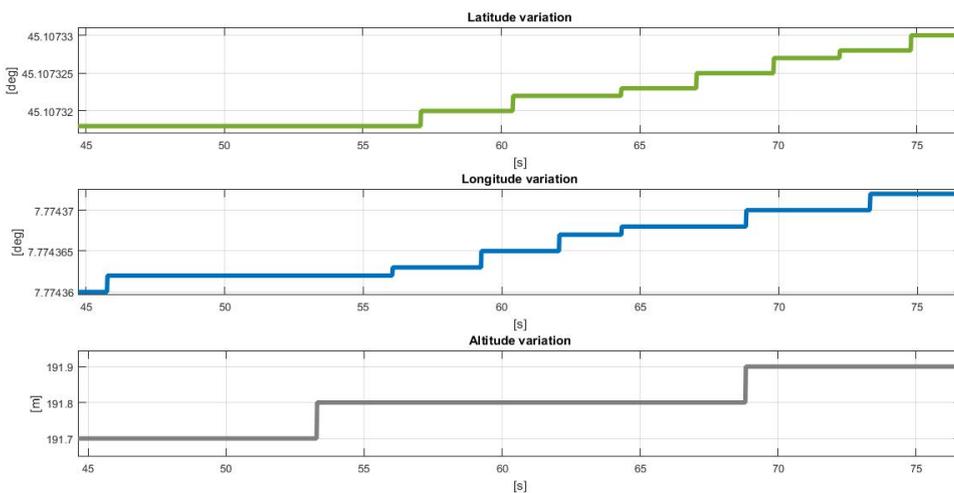
*Figura 5.10: Test all'aperto svolto il 26 agosto 2018. La sonda, lanciata da un piccolo colle, acquisisce velocità e si stabilizza. Viene manovrato in remoto, facendo qualche virata di 180° per poi tornare verso il punto di lancio. Non riuscendo ad acquisire quota, atterra in una zona intermedia (nell'immagine in basso a destra).*

- la longitudine proveniente dall'unità GPS
- l'altitudine proveniente dall'unità GPS
- angolo di imbardata (yaw) proveniente dal filtro di Madgwick
- angolo di beccheggio (pitch) proveniente dal filtro di Madgwick
- angolo di rollio (roll) proveniente dal filtro di Madgwick
- accelerazione lineare lungo l'asse longitudinale  $a_x$  dall'IMU come tutti quelli che seguono
- accelerazione lineare lungo l'asse laterale  $a_y$
- accelerazione lineare lungo l'asse direzionale  $a_z$
- velocità giroscopica attorno all'asse longitudinale  $g_x$
- velocità giroscopica attorno all'asse laterale  $g_y$
- velocità giroscopica attorno all'asse direzionale  $g_z$
- campo magnetico lungo l'asse longitudinale  $m_x$
- campo magnetico lungo l'asse laterale  $m_y$
- campo magnetico lungo l'asse direzionale  $m_z$

Abbiamo utilizzato uno script Matlab per creare i grafici dai dati provenienti dai sensori.



(a)



(b)

Figura 5.11: (a) Traiettoria in termini di latitudine, longitudine e altitudine. Si nota come i dati ricavati dall'unità GPS non sono sufficienti per tracciare una traiettoria dettagliata. Infatti, il GPS non riesce a discriminare una differenza di quota così piccola come nel nostro caso, dando dei risultati sbagliati. (b) Traiettoria in funzione del tempo. Anche qui si nota come l'interpolazione sia di poca utilità poiché i dati prelevati non variano, a causa del range di azione molto piccolo da parte del sistema. Soprattutto l'altitudine, che secondo l'unità GPS, varia di soltanto venti centimetri, quando in realtà vi sono qualche unità di metri. Quindi è sbagliato.

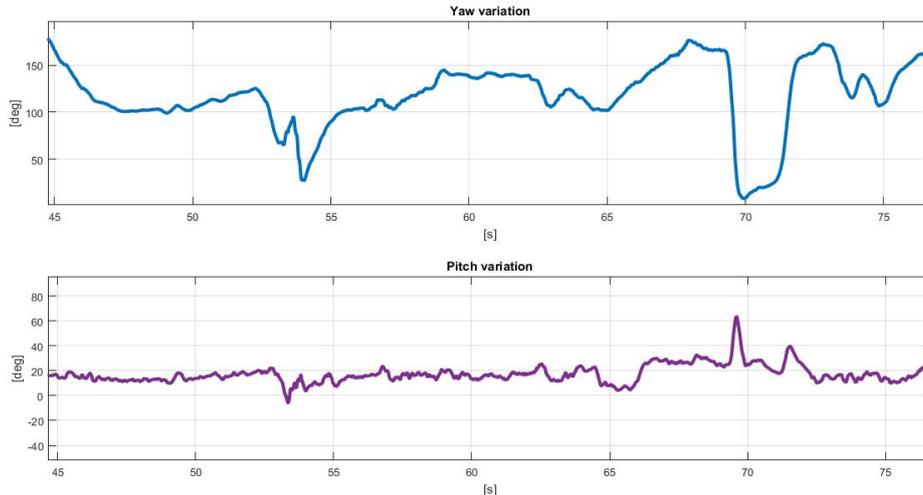


Figura 5.12: Andamento dell'angolo di imbardata (*yaw*) e beccheggio (*pitch*) nel tempo.

**Traiettoria** Il grafico della traiettoria è indicato nella figura 5.11 e si intuisce come l'unità GPS non sia in grado di fornire dati accurati poiché non riesce a discernere differenze di quote e di gradi così ridotti, come nel test eseguito. Infatti, per avere una traiettoria dettagliata e corretta è necessario eseguire il test con differenza di quote di decine, sennò centinaia di metri, e raggi di curvatura dello stesso ordine di grandezza. Infatti, qui viene mostrato come in questo tipo di test l'unità GPS non è di grande utilità. Esiste una possibilità per ricavare la traiettoria a partire dalle accelerazioni indicate dalla piattaforma inerziale. Il problema sta nel sapere con grande precisione le condizioni iniziali in termini di velocità.

**Assetto** Gli angoli di assetto indicati nella figura 5.12. L'angolo di imbardata dipende anche dal magnetometro della piattaforma inerziale e quindi il suo valore dipende da come è girata la piattaforma rispetto al Nord magnetico, mentre l'angolo di beccheggio e rollio (non indicato nella figura) dipendono solo dall'accelerometro e dal giroscopio. I loro valori, in caso di posizionamento orizzontale della piattaforma rispetto alla superficie terrestre saranno intorno a zero gradi.

**Accelerazioni** L'andamento delle accelerazioni lineari nel tempo sono indicate nella figura 5.13. Da notare come l'accelerazione  $a_z$  sia vicino al valore dell'accelerazione di gravità, mentre  $a_x$  e  $a_y$  abbiano un valore medio vicino allo zero. Da notare inoltre che quando vi è una variazione di accelerazione su un asse, ne risentono anche le altre due.

**Velocità giroscopiche** Per esaminare le velocità giroscopiche si osservi la figura 5.14. Prima che il sistema sia utilizzato, si ricorda, sono state eseguite delle calibrazioni sulle velocità giroscopiche, quindi è corretto che siano inizialmente nulle a sistema fermo.

**Campi magnetici** La variazione dei campi magnetici nel tempo è indicato nella figura 5.15.

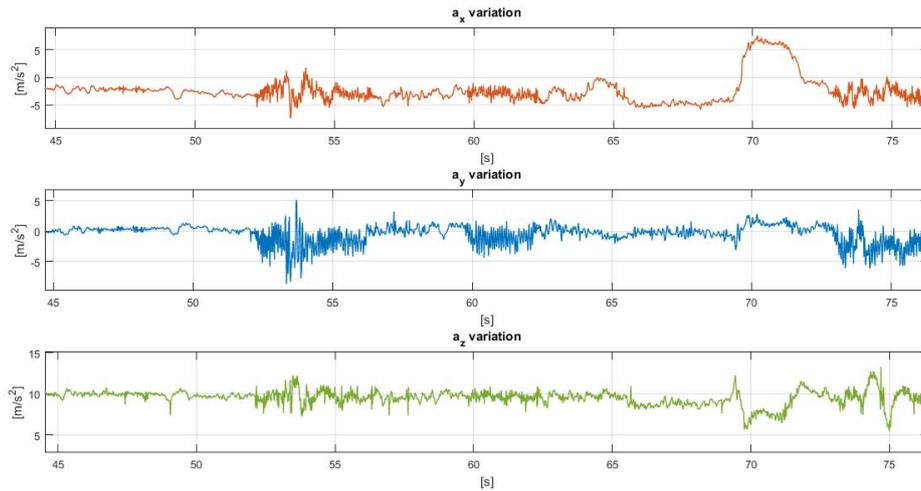


Figura 5.13: Andamento delle accelerazioni in funzione del tempo.

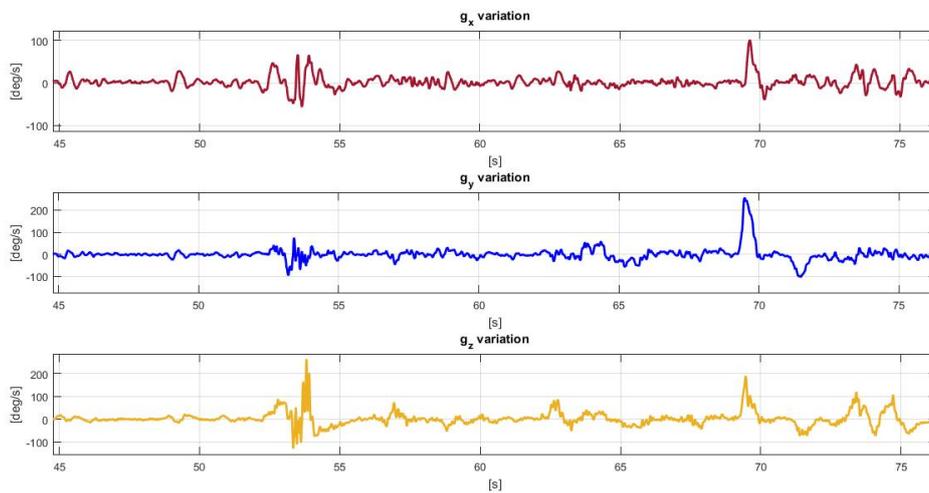


Figura 5.14: Andamento delle velocità giroscopiche in funzione del tempo.

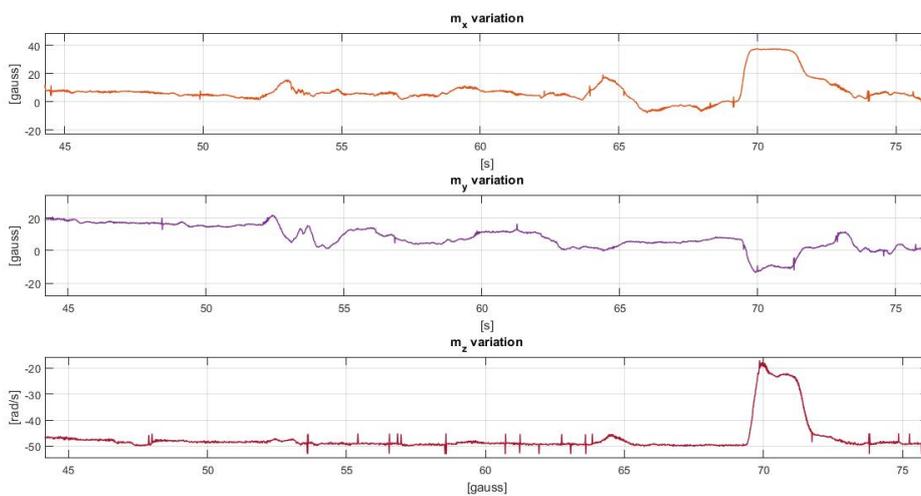


Figura 5.15: Andamento dei campi magnetici in funzione del tempo.



## 6 | Conclusioni

### 6.1 Raggiungimento degli obiettivi

L'idea iniziale era di sviluppare un sistema di rientro funzionante, controllato da un parafoil e da una serie di dispositivi elettronici e di riuscire a testare - in maniera preliminare - separatamente, con test al banco, e complessivamente, con test di volo, ogni aspetto di questo sistema. La maggior parte degli obiettivi proposti sono stati raggiunti. In particolare:

- ✓ Progettazione di una configurazione *ex novo* (sono state proposte due configurazioni diverse)
- ✓ Disegno CAD, stampa 3D e costruzione della struttura
- ✓ Integrazione della struttura con l'elettronica di bordo
- ✓ Test al banco su ogni componente elettronico presente a bordo
- ✓ Sviluppo di una logica di controllo remoto del sistema implementata su schede digitali
- ✓ Test di volo controllato con l'acquisizione di dati provenienti dai sensori

### 6.2 Criticità riscontrate

Vi sono state alcune criticità. La maggior parte sono state superate, altre, invece, richiedevano ulteriore tempo per approfondirle. Di questi eventuali studi futuri si parlerà nella sezione successiva [6.3](#).

**Peso** Il requisito sul peso, come già detto sin da subito, è stato il più stringente tra i requisiti, poiché si partiva da un parafoil già in possesso e un'idea di utilizzare un contenitore per parafoil già a disposizione. Il limite di 300 grammi era e, parzialmente, è tutt'ora un obiettivo da raggiungere. Il peso dell'elettronica di bordo supera, senza il peso del cablaggio, i 210 grammi. Quindi in 90 grammi rimanenti a disposizione è necessario includere il peso del cablaggio, delle strutture, delle protezioni e del materiale collante. Si è cercato di ottimizzare al massimo, soprattutto per la seconda configurazione, per arrivare, infine, a 320 grammi. Quindi sul requisito di sistema abbiamo sfiorato il peso del 6%. Questo esubero di peso porta con sé una serie di problemi durante la fase dell'utilizzo del parafoil.

**Protezione del hardware** Questa criticità è andata di pari passo con il controllo del peso complessivo. Una volta integrata l'elettronica di bordo con la struttura si doveva in qualche modo cercare di proteggere il hardware esposto. Abbiamo aggiunto quindi altre strutture e, nonostante fossero in polistirolo, abbiamo appesantito ancora di più il sistema. Una particolare attenzione è stata fatta nella zona in cui le ventole intubate venivano attaccate ai servomeccanismi che dovevano farle inclinare: infatti, era la zona più sensibile di tutto il sistema per come è stato progettato.

Dopo un primo test al banco sulla sollecitazione a flessione della struttura che doveva reggere sia la ventola che il servomeccanismo, si è aggiunta una contra-ventatura che l'ha irrigidita. Durante i test sulla prima configurazione, dopo un urto col suolo, l'attacco della ventola con il servo si è rotto, mentre la parte irrigidita con la contra-ventatura ha retto.

Infine, l'aggiunta degli aerofreni per la prima configurazione e dell'anello protettivo per la seconda configurazione ha permesso una efficiente protezione del hardware esposto.

**Incidenti durante i test** Vi sono stati alcuni incidenti durante i test, alcuni dei quali sono: la rottura dell'attacco della ventola intubata al servomeccanismo; l'ingarbugliarsi di un filo nel parafoil dentro un motore, con le conseguenze di dover tagliare e ricucire il filo; il distacco delle ventole durante i test. Tutti questi incidenti portano a riparazioni, revisioni e ricalibramenti.

## 6.3 Sviluppi futuri

Per mancanza di tempo alcuni aspetti non sono stati approfonditi in maniera sufficiente e quindi sono necessari alcuni ulteriori studi per il completamento del sistema di rientro. Questi sviluppi potranno e dovranno modificare parzialmente o pesantemente l'attuale configurazione.

### 6.3.1 Sistema di espulsione del Parafoil

Il sistema di espulsione dovrà essere il più leggero possibile e possibilmente senza l'aggiunta di un servomeccanismo dedicato. Eventualmente, un meccanismo semplice di levette, elastici o materiali innovativi, che sia in grado di sfruttare i quattro servomeccanismi già a bordo, potrebbe essere una soluzione valida.

### 6.3.2 Studio sulla spinta delle ventole

La spinta complessiva data dalle due ventole EDF-50 utilizzate qui è di poco superiore a 100 grammi. La velocità che acquisisce il sistema e il parafoil non sono molto alte. La portanza così sviluppata da quest'ultimo è anche essa piccola. Si potrebbe aumentare la spinta, aumentando così l'intensità della forza portante e quindi permettendo un volo in discesa con un angolo di rampa più piccolo e pertanto un volo più duraturo. Per fare ciò è necessario approfondire lo studio sulla spinta fornita dalle ventole e trovare, a scapito di una modifica della configurazione, un'alternativa migliore, che sia in grado di fornire una spinta maggiore a parità di peso e durata della batteria.

### **6.3.3 Test aggiuntivi all'esterno**

Per riuscire ad avere un'acquisizione di dati GPS migliore è necessario svolgere ulteriori test in un'area che dia la possibilità di lanciare il sistema da quote più alte, da cento metri ad esempio, e di poter effettuare manovre in un range ampio e a discesa lenta. In questa maniera si darà il tempo necessario ai sensori per funzionare in condizioni ottimali. I dati riguardante la quota sono quelli che hanno maggiore errore. Essi dovranno essere confrontati con dati di quota provenienti da altri sensori, come ad esempio un altimetro barometrico. I dati provenienti da sperimentazioni con dislivelli di quota più significativi darebbe la possibilità di svolgere un'analisi più approfondita sulla traiettoria.

### **6.3.4 Ulteriore sviluppo del software di controllo**

Si è sviluppato un software che permette di manovrare da tastiera, in remoto, tutto il hardware presente a bordo del sistema. Il passo successivo sarebbe quello di sviluppare un programma che, una volta ricevuti in input le coordinate del punto di atterraggio in termini di latitudine, longitudine e quota, più altri parametri di discesa, sia in grado di manovrare autonomamente i servomeccanismi e i motori per raggiungere il punto di atterraggio desiderato. Dentro il software vi è già una logica di controllo sull'angolo di imbardata, che permette al sistema di orientarsi nella direzione di un angolo di imbardata indicato. Sarebbe opportuno espandere tale logica anche alle altre grandezze in gioco.

### **6.3.5 Ottimizzazione del peso**

Con lo sviluppo del hardware, sarà possibile giocare sull'alleggerimento del peso dei dispositivi elettronici. Come ad esempio è stato fatto qui rispetto al lavoro precedente a questo [7]: infatti, qui si è utilizzato un Raspberry Pi Zero, sei volte più leggero del precedente Raspberry Pi B+ sei volte più pesante. Inoltre, è possibile uno sviluppo ulteriore anche nella parte delle strutture: una ri-progettazione completa oppure un utilizzo diverso di materiali con cui costruire le strutture stesse.



# Bibliografia

- [1] Ultimaker. *Ultimaker 2+: Engineered to perform*. 2018. URL: <https://ultimaker.com/en>.
- [2] Nobuyuki Yajima Naoki Izutsu Takeshi Imamura Toyoo Abe. *Scientific Ballooning - Technology and Applications of Exploration Balloons Floating in the Stratosphere and the Atmospheres of Other Planets*. A cura di Springer. 2009.
- [3] Adafruit. *Adafruit Ultimate GPS on the Raspberry Pi*. 2018. URL: <https://learn.adafruit.com/adafruit-ultimate-gps-on-the-raspberry-pi/setting-everything-up>.
- [4] David Grayson. *GitHub - DavidEGrayson/ahrs-visualizer: Draws a 3D representation of an IMU's orientation. Developed for Raspberry Pi and OpenGL ES. Works with minimu9-ahrs*. 2018. URL: <https://github.com/DavidEGrayson/ahrs-visualizer>.
- [5] PTC. *Software di modellazione 3D Creo Parametric | PTC*. 2018. URL: <https://www.ptc.com/it/products/cad/creo/parametric>.
- [6] Ultimaker. *Ultimaker Software - all platforms*. 2018. URL: <https://ultimaker.com/en/products/ultimaker-cura-software/list>.
- [7] Giovan Battista Errante. «Development of an Autonomous Parafoil recovery system for a high altitude stratospheric balloon». Tesi Laurea Magistrale. Politecnico di Torino, 2017.
- [8] T.A. Devezas F.A. d'Oliveira F.C.L. de Melo. «High-Altitude Platforms - Present Situation and Technology Trends». In: *Journal of Aerospace Technology and Management* (set. 2016). URL: [http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S2175-91462016000300249](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S2175-91462016000300249).
- [9] Eric S. Raymond. *GPSd - Put your GPS on the net!* 2017. URL: <http://www.catb.org/gpsd/>.
- [10] ICAO. *Annex 2 to the Convention on International Civil Aviation - Rules of the Air*. Catalogue of ICAO Publications, lug. 2005.
- [11] W. V. Jones J. A. Gaskin I. S. Smith. «Introduction to the special issue on scientific balloon capabilities and instrumentation». 2015. URL: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=2ahUKEwiK07v4zvvcAhXks4sKHwd1AW4QFjABegQICBAC&url=https%3A%2F%2Fnttrs.nasa.gov%2Farchive%2Fnasa%2Fcasi.nttrs.nasa.gov%2F20150023509.pdf&usg=A0vVawOyQSdeOR8sCxvli-PJ79eb>.
- [12] Collège des Ingénieurs J. Lòpez-Urdiales. «The Role of Balloons in the Future Development of Space Tourism». In: *53rd International Astronautical Congress* (ott. 2002).

- [13] Charles Curley Eric S. Raymond. *libgps - C service library for communicating with the GPS daemon*. 2017. URL: <http://www.catb.org/gpsd/libgps.html>.
- [14] x-io Technologies Limited. *Open source IMU and AHRS algorithms - x-io Technologies. Madgiwck Filter*. 2016. URL: <http://x-io.co.uk/open-source-imu-and-ahrs-algorithms/>.
- [15] Pololu Corporation. *Maestro Servo Controller Windows Drivers and Software (release 130422) (5MB zip)*. 2013. URL: <https://www.pololu.com/file/0J266/maestro-windows-130422.zip>.
- [16] Pololu Corporation. *Pololu Maestro Servo Controller User's Guide*. 2012. URL: <https://www.pololu.com/docs/pdf/0J40/maestro.pdf>.
- [17] David Grayson. *GitHub - DavidEGrayson/minimu9-ahrs: Program for reading data from the Pololu MinIMU-9 over I2C. Works on the Raspberry Pi and other embedded ARM Linux boards*. 2018. URL: <https://github.com/DavidEGrayson/minimu9-ahrs>.
- [18] Glenn Baddeley. *GPS - NMEA sentence information*. 2001. URL: <http://aprs.gids.nl/nmea/#refs>.
- [19] Raspberry Pi Community Manager Ben Nuttall. *Download NOOBS for Raspberry Pi*. 2018. URL: <https://www.raspberrypi.org/downloads/noobs/>.
- [20] Pololu Corporation. *Pololu - MinIMU-9 v3 Gyro, Accelerometer, and Compass (L3GD20H and LSM303D Carrier)*. 2018. URL: <https://www.pololu.com/product/2468>.
- [21] Pololu Corporation. *Pololu - Micro Maestro 6-Channel USB Servo Controller (Assembled)*. 2018. URL: <https://www.pololu.com/product/1350>.
- [22] Simon Tatham (PuTTY Team Member). *Download PuTTY: latest release*. 2018. URL: <https://www.chiark.greenend.org.uk/~sgtatham/putty/>.
- [23] SD Association. *SD Memory Card Formatter for Windows Download - SD Association*. 2018. URL: [https://www.sdcard.org/downloads/formatter\\_4/eula\\_windows](https://www.sdcard.org/downloads/formatter_4/eula_windows).
- [24] Ravi Vaidyanathan Sebastian O.H. Madgwick Andrew J.L. Harrison. «Estimation of IMU and MARG orientation using a gradient descent algorithm». In: *2011 IEEE International Conference on Rehabilitation Robotics* (giu. 2011).
- [25] Pololu Corporation. *Pololu - 5.h.1. Cross-platform C*. 2018. URL: <https://www.pololu.com/docs/0J40/5.h.1>.
- [26] Loon LLC. *Loon*. 2018. URL: <https://loon.co/>.
- [27] Wikipedia. *Facebook Aquila - Wikipedia*. 2018. URL: [https://en.wikipedia.org/wiki/Facebook\\_Aquila](https://en.wikipedia.org/wiki/Facebook_Aquila).
- [28] Dave Lee North America technology reporter for BBC. *Facebook abandons its Project Aquila flying internet plan - BBC News*. 2018. URL: <https://www.bbc.co.uk/news/technology-44624702>.
- [29] Japan Aerospace Exploration Agency. *ISAS / BALLOONS: Research on Balloons to Float Over 50km Altitude / Special Feature*. 2008. URL: <http://www.isas.jaxa.jp/e/special/2003/yamagami/03.shtml>.

- [30] Google. *Google Earth Pro per desktop - Google Earth*. 2018. URL: <https://www.google.com/intl/it/earth/desktop/>.
- [31] National Weather Service US Department of Commerce NOAA. *NWS JetStream - Radiosondes*. 2017. URL: <https://www.weather.gov/jetstream/radiosondes>.
- [32] World Meteorological Organization. *GOS Components*. 2018. URL: <http://www.wmo.int/pages/prog/www/OSY/Gos-components.html#upper>.
- [33] Thales Group. *What's up with Stratobus? Thales Group*. 2018. URL: <https://www.thalesgroup.com/en/worldwide/space/news/whats-stratobus>.
- [34] Zero 2 Infinity S.L. *Zero 2 Infinity, Providers of Access to Space. Bloon. Bloostar. Elevate*. 2016. URL: <http://www.zero2infinity.space/>.
- [35] Opale Paramodels. *Opale Paramodels - Oxy 0.5*. 2018. URL: <https://www.opale-paramodels.com/gb/rc-paramotor-paraglider-wing/37-oxy-05-3760245320116.html>.
- [36] Adafruit Industries. *Adafruit Ultimate GPS Breakout - 66 channel w/10 Hz updates - Version 3*. 2018. URL: <https://www.adafruit.com/product/746>.
- [37] STMicroelectronics. *L3GD20H - MEMS motion sensor: three-axis digital output gyroscope. Datasheet Production Data*. 2013. URL: <https://www.pololu.com/file/0J731/L3GD20H.pdf>.
- [38] STMicroelectronics. *LSM303D Ultra compact high performance e-Compass 3D accelerometer and 3D magnetometer module. Datasheet - preliminary data*. 2012. URL: <https://www.pololu.com/file/0J703/LSM303D.pdf>.
- [39] Ultimaker. *Professional 3D printing made accessible | Ultimaker*. 2018. URL: <https://ultimaker.com/en>.
- [40] Wiring Pi. *Raspberry Pi Wiring, Download and Install Wiring Pi*. 2018. URL: <http://wiringpi.com/download-and-install/>.
- [41] WiringPi. *I2C Library | Wiring Pi*. 2018. URL: <http://wiringpi.com/reference/i2c-library/>.