



UNIVERSITÀ POLITECNICA DELLE MARCHE

Corso di Laurea Triennale in
Ingegneria Informatica e dell'Automazione

Studio della catena Batteria-ESC-Motore- Brushless-Elica di un Drone

Study of the Battery-ESC-Motor-Brushless-
Propeller Chain of a Drone

Docente:
Prof. Andrea Bonci

Studente:
Lena Marison

A.A. 2022/2023

INDICE

INTRODUZIONE	5
1 Cos'è un Quadricottero	5
1.1 Cenni Storici.....	5
1.2 Fisica di Quadricottero	6
1.3 Modello matematico quadricottero	7
STUDIO INIZIALE	9
2 Relazione tra i componenti.....	9
2.1 Batteria	9
2.2 ESC.....	9
2.3 Motore Brushless.....	11
2.4 Eliche.....	12
3 Relazione di tutto l'anello.....	13
3.1 Relazione tra tensione, corrente e potenza per la batteria:.....	13
3.2 Relazione tra il segnale PWM e la velocità dell'ESC:	13
3.3 Relazione tra il numero di giri del motore brushless e la tensione applicata:	14
3.4 Relazione tra il passo delle eliche e la spinta generata:	14
HARDWARE E SOFTWARE.....	15
4 HARDWARE.....	15
4.1 Scheda STM32H745 Nucleo-144.....	15
4.2 Batteria LiPo Turnigy A-Spec 2600 mAh 4S.....	16
4.3 ESC Turnigy Plush 30A	17
4.3.1 Programmazione dell'ESC.....	17
4.3.2 Procedura di calibrazione.....	18
4.3.3 Procedura di armamento	19
4.4 Motore Brushless Turnigy D3536/9 910KV.....	20
4.5 Eliche Slow Fly Electric Prop 9057SF 10x5	20
4.6 Encoder ATM10	21
5 SOFTWARE.....	23
5.1 STM32CubeIDE.....	23
5.2 Serial Tools.....	23
5.3 MATLAB	24
SVILUPPO DEL CODICE	25
6 Configurazione periferiche.....	25
6.1 PWM	25
6.2 ADC.....	27
6.3 Time Encoder mode.....	27
7 Codice.....	28
7.1 Trimmer per la modifica del duty cycle.....	28
7.2 Filtro utilizzato per il trimmer	29
7.3 Calibrazione dell'ESC	30
7.4 Armatura dei motori	31
7.5 Codice complessivo per test del motore	31
7.6 Codice Lettura Encoder.....	32
7.7 Codice con incremento dinamico del duty cycle	33
7.8 Codice Matlab	33
STUDIO FINALE	37
8 Ottimizzazione del Controllo del Motore Brushless	37

8.1	Calibrazione e Armamento (setup 1).....	38
8.2	Range di duty cycle per la calibrazione (setup1).....	39
8.3	Effetto della modalità di avvio dell'ESC.....	41
8.4	Replicazione dei test su setup alternativo (setup2).....	41
8.5	Misura della velocità tramite encoder.....	42
8.6	Studio del legame duty cycle- velocità (RPM) tramite l'encoder.....	44
9	Test e Grafici	45
9.1	Studio dei limiti del brushless.....	49
	<i>CONCLUSIONI E SVILUPPI FUTURI.....</i>	51
	<i>Bibliografia</i>	55

INTRODUZIONE

L'obiettivo principale di questa relazione è analizzare il funzionamento del sistema di propulsione di un drone quadricotore ed in particolare l'interconnessione dei suoi quattro componenti fondamentali: batteria, ESC (Electronic Speed Controller), motore brushless ed eliche. È infatti necessario comprendere come queste componenti interagiscano tra loro nel determinare il comportamento complessivo del sistema per poter ottenere un controllo affidabile del quadricotore.

Nello specifico, il lavoro svolto ha previsto una fase iniziale di studio delle singole componenti e delle loro relazioni e una successiva fase di sperimentazione sul sistema di propulsione di un quadricotore presente in laboratorio per verificarne l'effettivo comportamento.

1 Cos'è un Quadricottero

1.1 Cenni Storici

Nel corso del tempo si sono sviluppate varie tipologie di velivoli aerei, tra cui: veicoli a rotore centrale verticale, come gli elicotteri, con o senza rotore di coda (ovvero quel dispositivo che consente di annullare il momento torcente dovuto alla rotazione della pala centrale, in quest'ultimo caso, si adoperava un secondo rotore, che ruota in modo opposto al primo e si parla di doppio rotore, a configurazione coassiale); velivoli a rotore centrale orizzontale, come lo sono i vecchi aeroplani; velivoli birotori come possono essere i veicoli sofisticati in dotazione agli eserciti militari; velivoli multi-rotore che vanno da tre a otto rotori.

I quadricotteri rientrano in quest'ultima categoria, essendo dei sistemi a quattro rotori disposti a croce (Figura 1), e oggi giorno sono tra i velivoli più popolari. Il quadricotore fu per la prima volta realizzato come veicolo per il trasporto di passeggeri, ma ben presto uscì di scena a causa del peso eccessivo dovuto alle strutture necessarie a sostenere i 4 rotori ed alla scarsa stabilità che gravava sulla manovrabilità del pilota. Recentemente, con il diffondersi di veicoli denominati UAV (Unmanned Aerial Vehicle), questi sistemi hanno trovato svariate applicazioni in ambito civile tra cui: la didattica, l'hobbismo e l'esplorazione di ambienti impervi o pericolosi per gli esseri umani. I sistemi UAV, introdotti in ambito militare per sorvegliare aree pericolose o inaccessibili a causa di condizioni ambientali avverse, utilizzano sistemi di controllo computerizzati che mediante l'elaborazione dei dati provenienti da sensori di posizione e orientamento sono in grado di pilotare il velivolo autonomamente o di rendere possibile la guida da terra mediante radiocomando. Le potenzialità e i vantaggi di questi mini-veicoli derivano dalla loro manovrabilità, dalla possibilità di operare in situazione di hovering, (ovvero praticamente immobili), "autobilanciati" dal sistema di controllo autonomo. Per di più, se dotati di un controllo abbastanza efficace, possono essere utilizzati anche all'interno di un ambiente chiuso, e quindi, si rendono a tutti gli effetti disponibili per i più disparati servizi.



Figura 1: Drone quadricottero

1.2 Fisica di Quadricottero

Fisicamente, un quadricottero si presenta come un corpo rigido simmetrico (frame), formato da quattro aste orizzontali che montano alle estremità quattro rotori a pala fissa e opportunamente dimensionati. La simmetria del dispositivo consente di centralizzare tutto il carico sollevabile (payload), massimizzato rispetto ad un tradizionale elicottero, al centro del frame. La dinamica del movimento è completamente attuata agendo solo sui contributi di ogni singolo motore e, ulteriore differenza con i comuni velivoli aerei, le pale dei quattro rotori ruotano a due a due in senso contrario e sono di tipo normale (sinistrorso), per i motori 2-4 e, inverso (destrorso), per i motori 1-3. Questa caratteristica consente di annullare il momento angolare dovuto agli effetti di rotazione delle pale e, quindi, di annullare l'angolo di imbardata (yaw) quando tutti i motori girano con la stessa velocità (Figura 2).

Nella medesima situazione, mediante relazioni matematiche è possibile aumentare o diminuire le velocità dei motori, che sono strettamente collegate alla portanza delle pale (f) ed alla forza totale di spinta (u). Tramite ciò, è possibile controllare il moto del velivolo lungo l'asse z , ovvero: l'altitudine.

Per quanto riguarda, invece, i movimenti lungo gli assi orizzontali, essi sono controllati agendo sulle coppie di rotori simmetrici, mantenendo la rotazione dei restanti alla stessa velocità. In riferimento alla (Figura 2(b)), vediamo che un aumento della velocità del motore 1 e una corrispettiva diminuzione della velocità del motore 3 (o viceversa) implica una rotazione attorno all'asse trasversale, generando un angolo di beccheggio (pitch). Allo stesso modo, creando invece uno scompensamento tra le velocità dei motori 2 e 4 si genera l'angolo di rollio (roll), come si può evincere dalla (Figura 2(c)).

In basso a destra (Figura 2(d)) è invece rappresentata la configurazione adibita alla rotazione sul piano x, y . In questa configurazione, due motori opposti (es. 2-4) diminuiscono la loro velocità rispetto all'altra coppia di motori (es. 1-3), generando complessivamente un momento torcente diverso da zero e quindi una rotazione attorno all'asse z , che viene definita imbardata (yaw).

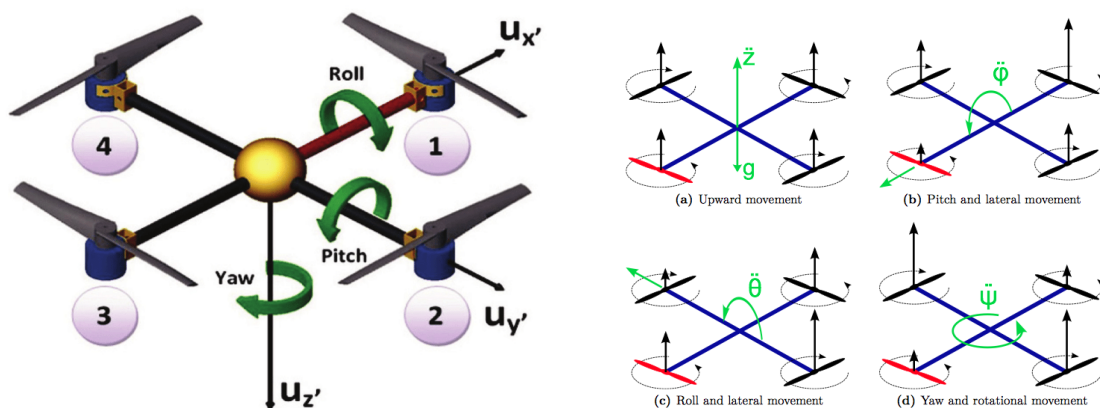


Figura 2: Movimenti del drone che si ottengono modificando la velocità di determinati motori

1.3 Modello matematico quadricottero

Prima di introdurre il modello matematico del drone, definiamo in Figura 3 i sistemi di riferimento su cui baseremo la trattazione:

$$E_I = [x, y, z]; E_B = [x_B, y_B, z_B]$$

dove:

E_I sistema di riferimento inerziale assoluto (suolo)

E_B sistema di riferimento solidale al corpo del drone (la cui origine è posta nel centro di massa del corpo)

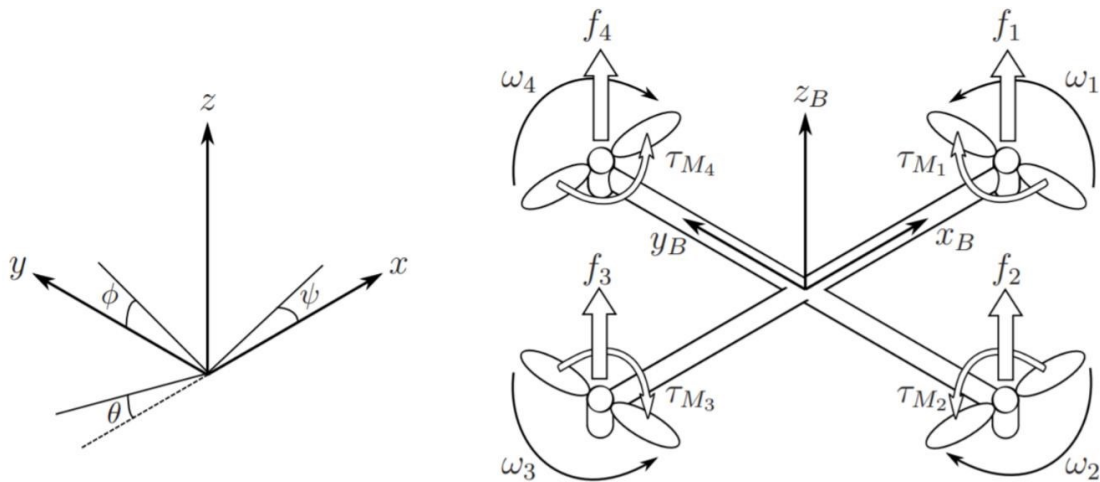


Figura 3: Rappresentazioni del sistema inerziale assoluto e di quello solidale al quadricottero

Inoltre, definiamo ξ la posizione assoluta del drone in E_I e η l'assetto del drone in E_I .

Osservando l'assetto più nel dettaglio abbiamo: l'angolo di roll ϕ che rappresenta la rotazione del quadricottero attorno all'asse x, l'angolo di pitch θ che rappresenta una rotazione attorno all'asse y e l'angolo di yaw ψ che rappresenta una rotazione attorno all'asse z.

$$\xi = \begin{bmatrix} x \\ y \\ z \end{bmatrix}; \quad \eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$$

Ora possiamo introdurre le velocità lineari V_B e angolari v viste dal sistema E_B :

$$V_B = \begin{bmatrix} v_{x,B} \\ v_{y,B} \\ v_{z,B} \end{bmatrix}; \quad v = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Possiamo ricondurre queste velocità alle corrispondenti $\dot{\xi}$ e $\dot{\eta}$, rispettivamente, attraverso la matrice di rotazione R e la matrice W_η :

$$R = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \cos \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi - \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \sin \phi \cos \phi \end{bmatrix}$$

$$W_\eta = \begin{bmatrix} 1 & \sin \psi \cos \phi & \cos \phi \tan \theta \\ 0 & \cos \phi & \sin \phi \\ 0 & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}$$

infatti:

$$\xi = RV_B; \quad \eta = W_\eta v;$$

Infine, si suppone che il drone sia perfettamente simmetrico, rispetto agli assi x e y. In questo modo gli assi principali d'inerzia coincidono con gli assi di simmetria e abbiamo che $I_{xx} = I_{yy}$.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

STUDIO INIZIALE

2 Relazione tra i componenti

La scelta accurata dei componenti è cruciale per garantire prestazioni ottimali e una durata prolungata. Questa relazione esplorerà la sinergia tra quattro componenti fondamentali: batteria, ESC, motore brushless ed elica. Questi quattro elementi, quando combinati, creano un sistema elettrico che influisce significativamente sulle prestazioni e sulla durata del drone o quadrirotore. Nello specifico in questo capitolo viene descritto il funzionamento dei singoli componenti e le relazioni funzionali e matematiche che intercorrono tra di essi.

Queste considerazioni generali verranno poi rese specifiche nel capitolo successivo, applicandole agli specifici componenti della catena utilizzata in fase di test che si anticipano essere: la batteria Turnigy A-Spec 2600mAh, l'ESC Turnigy Plush 30A, il Motore Brushless Turnigy D3536/9 910KV e le Eliche Bipala Slow Fly Electric Prop 9057SF da 10x5.

2.1 Batteria

La batteria è il cuore energetico del sistema. La sua capacità, espressa in milliampere-ora (mAh), determina quanto tempo il veicolo può operare prima di richiedere una ricarica. Le batterie più frequentemente utilizzate nei droni sono le LiPo (litio-polimero) in quanto offrono una potenza affidabile e una densità energetica elevata, dando al veicolo l'energia necessaria per spingere al massimo le prestazioni. Nella scelta della batteria è fondamentale verificare la sua compatibilità con l'ESC utilizzata per garantire una distribuzione stabile della potenza.

2.2 ESC

L'ESC è il componente responsabile di regolare la velocità del motore del veicolo. Il termine ESC deriva dall'acronimo inglese Electronic Speed Control (Controllore Elettronico di Velocità) ed è appunto quell'insieme di microcontrollore e banco di transistor atto a controllare le commutazioni delle fasi che permettono di regolare le performance di un motore brushless (Figura 4). Allo stesso tempo, essi provvedono all'alimentazione sia del motore sia eventualmente degli altri apparati elettronici del sistema (ad esempio la scheda a microcontrollore che controlla il sistema).

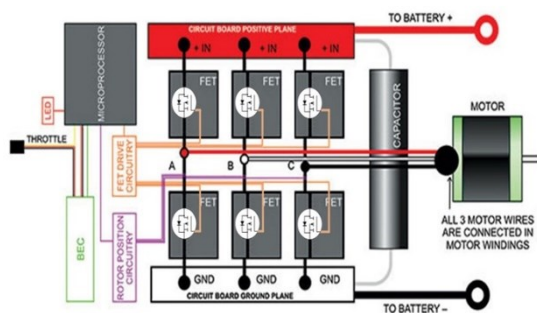


Figura 4: (a) ESC internal Circuit

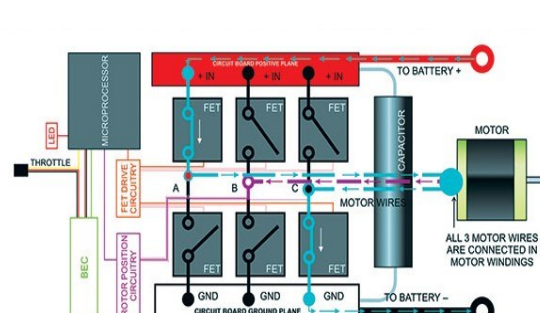


Figura 4: (b) Current flowing through ESC

L'ESC è stato introdotto per la prima volta nel campo del radio-modellismo e, proprio per questo, è stato realizzato appositamente per permettere di controllare un motore brushless attraverso un segnale PWM in modo del tutto analogo a come si controlla un servomotore. Un servomotore non è altro che un motore con driver integrato che, ricevendo un comando PWM da una ricevente radio o direttamente da una scheda a microcontrollore, è in grado di muoversi in determinate posizioni o raggiungere determinate velocità imposte dal trasmettitore. Nello specifico il controllo di un servomotore prevede di utilizzare un segnale PWM con frequenza di 50 Hz (periodo di 20ms) e un duty cycle che può variare da un minimo del 5% (1ms di periodo alto) a un massimo del 10% (2ms di periodo alto). Quando il servomotore riceve un segnale PWM con duty cycle minimo, esso si posiziona su un angolo di 0°, mentre con il duty cycle massimo raggiunge l'angolo massimo di 180°. Le ESC sono progettate per essere pilotate da un segnale PWM analogo sempre con frequenza di 50Hz e duty cycle nel range [5%-10%]. In questo caso il duty cycle minimo richiede all'ESC di alimentare le fasi del motore in modo da farlo ruotare alla velocità minima, mentre con il duty cycle massimo si ottiene la massima velocità. In Figura 5 è riportato il confronto tra i segnali di pilotaggio di un servomotore e di un brushless controllato tramite ESC.

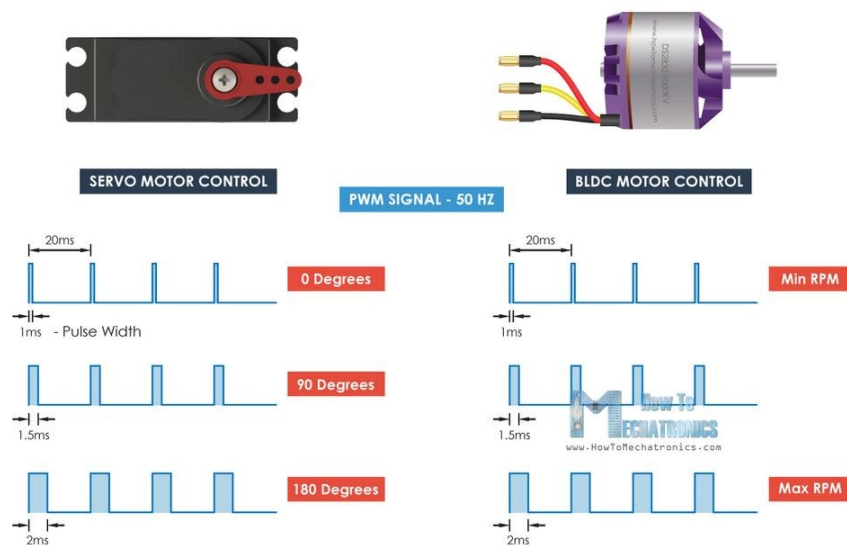


Figura 5: Confronto PWM per controllo servomotore e brushless

Le ESC sono dotate di un connettore standardizzato a 3 cavi, uno di essi (giallo/arancione) è utilizzato per il segnale PWM con cui la ricevente radio/ la scheda di controllo comanda la velocità del motore. Gli altri 2 cavi (nero e rosso) costituiscono invece il BEC (Battery Eliminator Circuit), ovvero circuito elimina batteria, che permette di utilizzare l'ESC per alimentare i componenti che vi sono connessi (ad esempio una ricevente radio o una scheda a microcontrollore), fornendo loro una tensione continua e stabilizzata a 5V ricavata dalla batteria centrale che alimenta i motori. Anche questo aspetto è pensato appositamente per le applicazioni aereo-modellistiche in cui le problematiche di peso e dimensioni non sono trascurabili.

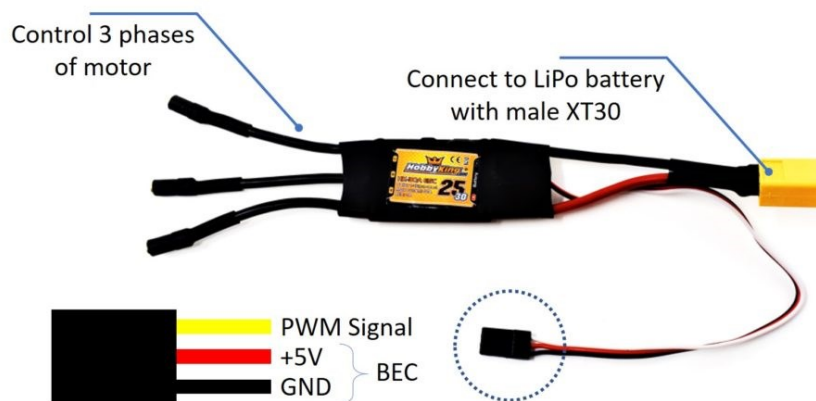


Figura 6: Cablaggio dell'ESC

2.3 Motore Brushless

Il Motore è il componente responsabile di convertire l'energia elettrica in movimento meccanico.

La scelta di utilizzare motori brushless, tipica nelle applicazioni di aeromodellismo, al posto dei più tradizionali motori a collettore è dovuta sostanzialmente a motivi di sicurezza (assenza di strofinio delle spazzole), maggior durata e migliori prestazioni in termini di velocità massima di rotazione e minor peso. Nello specifico nei droni vengono spesso utilizzati brushless a tre fasi, “outrunner” (a cassa rotante) e sensorless. Essendo dei motori “outrunner” hanno lo statore fisso, posto internamente insieme agli avvolgimenti delle tre fasi, ed il rotore esterno, costituito da magneti permanenti, che ruota attorno allo statore. Il vantaggio di utilizzare questa configurazione è l'elevata coppia sostenibile. Essendo sensorless, non sono dotati di sensoristica (come encoder o sensori magnetici) per determinare la posizione angolare del rotore bensì utilizzano un feedback proveniente dalla f.e.m. generatasi nel motore per determinare tale posizione e quindi la commutazione delle fasi che deve avvenire.

Il parametro fondamentale di un motore brushless è il suo KV, ovvero il coefficiente di velocità specifico del motore. Questo valore rappresenta il numero di giri al minuto che il motore raggiunge quando viene applicata una tensione di 1 volt.

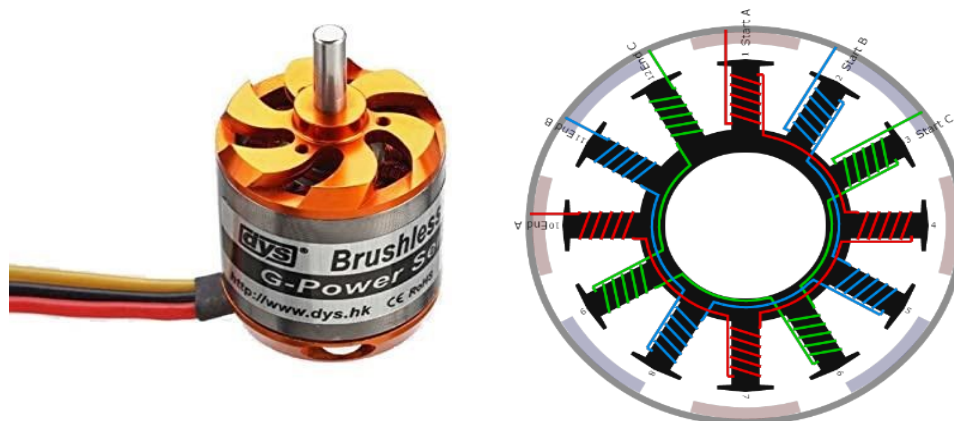


Figura 7: Esempio di motore brushless a 3 fasi outrunner sensorless

La commutazione delle fasi è gestita dall'ESC, che, sulla base del duty cycle del segnale PWM ricevuto e del feedback sulla posizione del rotore, pilota in maniera opportuna un bando di transistor per commutare ciclicamente le tre fasi e far quindi ruotare il motore.

L'ESC e il Motore brushless devono quindi essere scelti in modo che siano in grado di lavorare in perfetta sintonia. L'ESC fornisce un controllo preciso della velocità, consentendo al motore di rispondere istantaneamente alle richieste del pilotaggio. Il motore, con la sua potenza e coppia, offre la spinta necessaria per il veicolo, consentendogli di raggiungere alte velocità e affrontare terreni impegnativi. Questa sintonia perfetta tra ESC e motore crea un sistema altamente efficiente che massimizza le prestazioni dell'UAV.

2.4 Eliche

L'elica è il componente che consente di trasformare la velocità di rotazione del motore in spinta per il drone. L'equilibrio tra il motore e le eliche è fondamentale per garantire prestazioni ottimali e una gestione efficiente della spinta. Questa relazione esplorerà l'importante relazione tra il motore e l'elica. Questi due componenti, quando combinati in modo adeguato, creano un sistema che influenza significativamente le prestazioni e la maneggevolezza del veicolo.

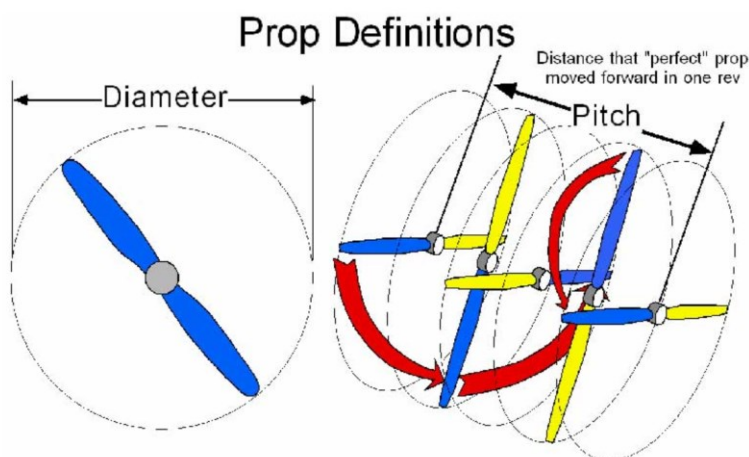


Figura 9: Parametri dell'elica

3 Relazione di tutto l'anello

Nel corso di questa tesina, esploreremo come i quattro componenti chiave - Batteria, ESC, Motore Brushless e Eliche - lavorino insieme per massimizzare le prestazioni negli UAV e nello specifico in un quadricottero. Oltre a esaminare le caratteristiche individuali di questi componenti, cercheremo di stabilire relazioni matematiche che descrivano la loro interazione e il loro impatto sulle prestazioni complessive.

3.1 Relazione tra tensione, corrente e potenza per la batteria:

Una delle prime relazioni da considerare è tra la batteria e la potenza fornita al sistema. Questa relazione può essere espressa dalla formula:

$$P = VI$$

Se si considera ad esempio una batteria a 4 celle con capacità di 2600mAh e scarica di 25C si ottiene:

$$P(W)(3,7 \times 4)(2.6A \times 25C) = 962W$$

La batteria fornisce la tensione necessaria per alimentare il sistema e la corrente necessaria per far funzionare il motore brushless attraverso l'ESC.

3.2 Relazione tra il segnale PWM e la velocità dell'ESC:

Per controllare la velocità del motore brushless si utilizza un ESC che risponde a un segnale PWM (Pulse Width Modulation). La relazione esatta tra il segnale PWM e la velocità del motore dipenderà dalla curva di calibrazione specifica dell'ESC.

Il PWM è una tecnica di modulazione in cui un segnale elettrico viene generato con impulsi periodici a una frequenza costante ma di larghezza variabile. Ogni impulso è caratterizzato da una "larghezza dell'impulso", che rappresenta la durata dell'impulso in rapporto al periodo totale dell'onda.

La larghezza dell'impulso varia per controllare la potenza consegnata a un dispositivo. Quando la larghezza dell'impulso è maggiore, il dispositivo riceve più potenza; quando è minore, il dispositivo riceve meno potenza.

Questa tecnica viene chiamata modulazione di larghezza d'impulso e permette di controllare in modo preciso la potenza consegnata al dispositivo mantenendo costante la frequenza del segnale.

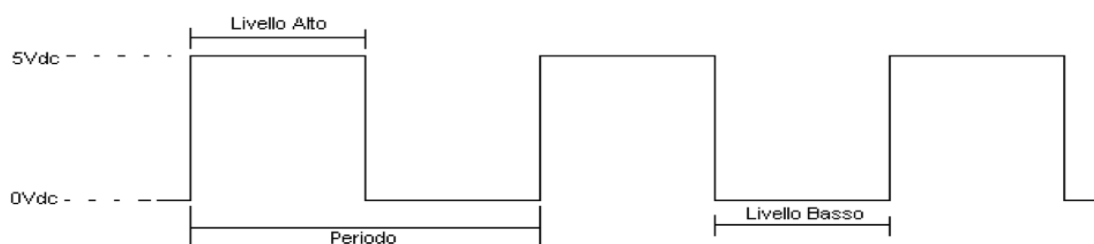


Figura 10: Esempio di segnale PWM

Il duty cycle è il rapporto che c'è tra il segnale positivo e il periodo totale del segnale ad onda quadra ed è espresso in percentuale. Un duty cycle del 100% equivale ad un segnale alto continuo, uno del 0% equivale ad un segnale basso continuo, mentre un duty cycle del 50% significa che il segnale alto dura quanto quello basso.

$$\text{Duty Cycle (\%)} = \frac{\text{Livello Alto}}{\text{Periodo}} \cdot 100$$

Dove 'Livello Alto' è il valore di tempo della durata del segnale positivo mentre 'Periodo' è il tempo relativo al periodo del segnale ad onda quadra. Per tenere il motore sempre con una certa forza a diverse velocità, non bisognava abbassare il valore di tensione, ma pilotare il motore con un segnale ad onda quadra e variare il duty cycle dello stesso.

3.3 Relazione tra il numero di giri del motore brushless e la tensione applicata:

Il Motore Brushless, con la sua classificazione di KV, risponde alla tensione applicata. La velocità del motore (RPM) può essere approssimativamente espressa come:

$$Velocità [RPM] = KV_{motore} \times Tensione[V]$$

- * 'Velocità' rappresenta il numero di giri del motore al minuto (RPM).
 - * 'KV_{motore}' è il coefficiente di velocità specifico del motore; questo valore rappresenta il numero di giri al minuto che il motore raggiunge quando viene applicata una tensione di 1 Volt.
 - * 'Tensione' è la tensione applicata al motore in Volt (V).
- Questo ci permette di comprendere come il motore risponda alle variazioni nella tensione fornita dalla batteria.

Se si considera ad esempio un motore con 910KV alimentato attraverso una batteria LiPo a 4 celle si ottiene:

$$V_{Max} [RPM] = (910 \times 15,8) = 14378 RPM$$

3.4 Relazione tra il passo delle eliche e la spinta generata:

L'elica svolge un ruolo cruciale nella generazione di spinta. La spinta generata può essere approssimativamente correlata al passo dell'elica attraverso una formula del tipo:

$$F = K(P_{elica})^2 \times D_{elica}^2$$

- * 'F' rappresenta la forza di spinta generata dalle eliche (indicata dal venditore).
 - * 'K' è un coefficiente che dipende da vari fattori, tra cui la forma dell'elica e le caratteristiche del flusso d'aria.
 - * 'P_{elica}' indica il passo dell'elica.
 - * 'D_{elica}' indica il diametro dell'elica.
- Questo ci aiuta a capire come le dimensioni delle eliche influenzino la capacità del sistema di propulsione del quadrirotore di generare spinta.

HARDWARE E SOFTWARE

4 HARDWARE

In questo capitolo, esploreremo i componenti hardware chiave utilizzati in fase di test, analizzandone le caratteristiche e il ruolo fondamentale che svolgono nel funzionamento complessivo del sistema.

4.1 Scheda STM32H745 Nucleo-144

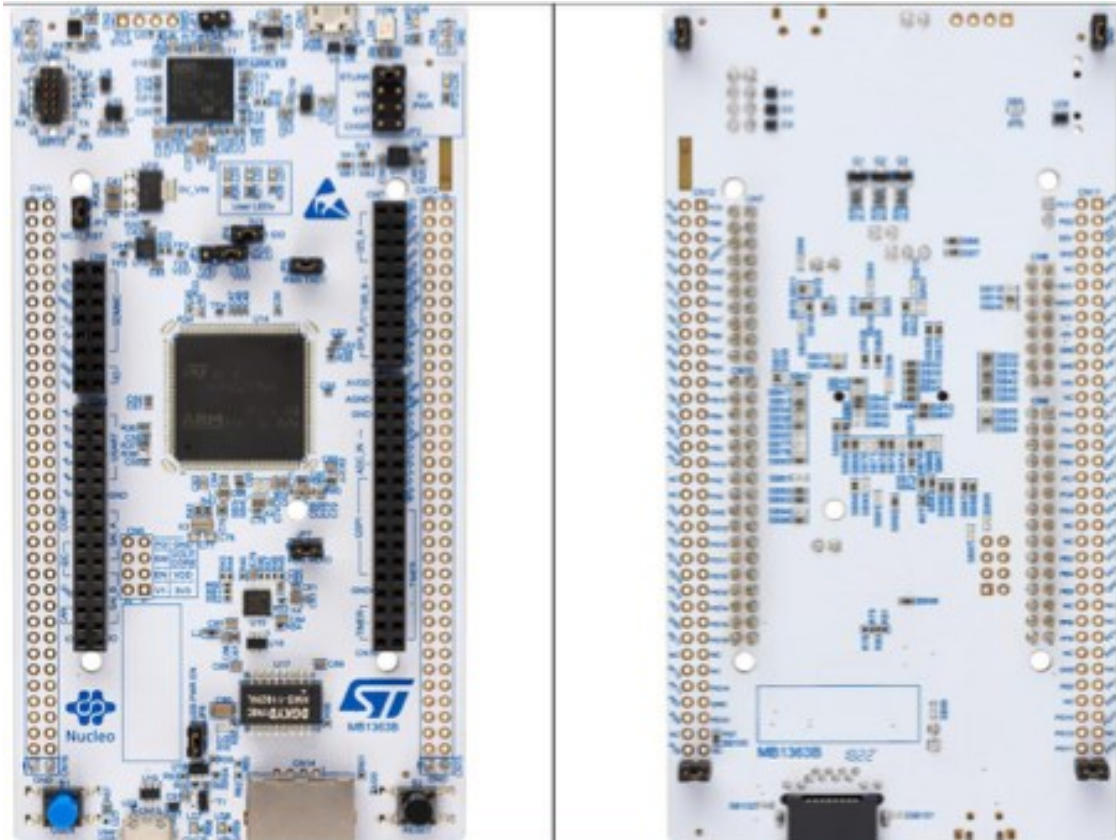


Figura 11: Vista fronte.

Vista retro.

Per il progetto è stata utilizzata la scheda NUCLEO H745ZI-Q che combina le prestazioni dell'Arm Cortex-M7 (con unità a virgola mobile a doppia precisione) fino a 480 MHz e del core Arm Cortex-M4 (con unità a virgola mobile a precisione singola) fino a 240 MHz.

Nel progetto è stato utilizzato solo il core Arm Cortex-M4.

Prestazioni

- FCPU a 480 MHz sul Cortex-M7 e 240 MHz sul Cortex-M4, 3224 CoreMark / 1327 DMIPS in esecuzione dalla memoria Flash, con stati di attesa 0 grazie alla sua cache L1.
- Cache L1 (16 Kbyte di I-cache + 16 Kbyte di D-cache) che migliora le prestazioni di esecuzione da memorie esterne.

Sicurezza

Gli MCU STM32H755 includono le seguenti funzionalità di sicurezza aggiuntive:

- Accelerazione hardware crypto/hash
- Servizi di sicurezza incorporati SFI (Secure Firmware Install) per autenticare e proteggere gli IP software durante l'esecuzione della programmazione iniziale.
- Aggiornamento del firmware sicuro all'avvio protetto (SBSFU)

Efficienza energetica

- L'architettura multi-power domain consente di impostare diversi domini di potenza in modalità a basso consumo per ottimizzare l'efficienza energetica.
- SMPS integrato per ridurre la tensione di alimentazione. Può anche essere utilizzato per fornire circuiti esterni e può anche essere combinato con l'LDO per casi d'uso specifici.
- Regolatore USB per alimentare lo strato fisico incorporato (PHY).
- 145 μ /MHz @VDD tipici = 3,3 V e 25 °C in modalità Run (periferiche spente) e SMPS
- 2,43 μ A tipico in modalità Standby (modalità a basso consumo)

Periferiche integrate

- Fino a 35 interfacce di comunicazione tra cui FD-CAN, USB 2.0 ad alta velocità/full-speed, Ethernet MAC, interfaccia telecamera
- Gamma di memoria facilmente estendibile utilizzando il controller di memoria flessibile con un'interfaccia parallela a 32 bit o l'interfaccia di memoria Flash seriale Quad-SPI dual- mode
- Analogico: DAC a 12 bit, ADC veloci a 16 bit
- Timer multipli a 16 e 32 bit in esecuzione fino a 480 MHz sul timer ad alta risoluzione a 16 bit

4.2 Batteria LiPo Turnigy A-Spec 2600 mAh 4S

Nel progetto è stata utilizzata per l'alimentazione una batteria Turnigy A-Spec 2600 mAh, 4S 25-50 Li-Po Pack. Di seguito sono elencate le caratteristiche:

Capacità	2600 mAh
Voltaggio	4S1P – 4 Celle – 14.8V
Scarica	25C costanti / 65C picco
Dimensioni	106x35x24mm
Peso	201g



Figura 12: Batteria Turnigy A-Spec 2600 mAh

La batteria ha una capacità di 2600(mAh)

La notazione "4S" indica che la batteria è composta da quattro celle Li-Po collegate in serie. Nella notazione "4S," "S" sta per "serie".

La batteria fornisce una tensione nominale di 14,8 Volt (4 celle Li-Po da circa 3,7 Volt ciascuna in serie), quando è totalmente carica la tensione erogata è di 16,8V (4 celle da 4.2V ciascuna).

La batteria è in grado di gestire correnti di scarica elevate, con una capacità di scarica continua compresa tra 25C e 65C. Questo significa che può erogare correnti più elevate senza surriscaldarsi o danneggiarsi.

4.3 ESC Turnigy Plush 30A

Nel progetto è stata utilizzata l'ESC Turnigy Plush 30A, di seguito le sue specifiche:

Tensione di alimentazione	14,6V
Tensione minima di funzionamento	5,6V
Corrente di lavoro	30°
BEC	5V 2A ver.3,1
Fonte Manuale	Turnigy Plush 30A



Figura 13: ESC Turnigy Plush 30A

4.3.1 Programmazione dell'ESC

Tali dispositivi possono essere programmati mediante una scheda di programmazione apposita che ne permette il settaggio di caratteristiche quali:

- **Freno (On/Off):** rappresenta la modalità di frenata dei motori. Quando è impostata su “On”, non appena l’ESC riceve il duty cycle minimo (che corrisponde ai motori fermi) comanda il motore in modo da frenarlo e farlo fermare immediatamente, altrimenti il motore rallenta naturalmente ed impiega quindi del tempo ad arrestarsi a causa della sua inerzia.
- **Tipologia di Cut-Off (Soft-cut/Cut-off):** consente di ridurre immediatamente la potenza fornita al motore una volta raggiunta la tensione di cut-off (vedi punto successivo) se settato su “Cut-off”, altrimenti la riduzione di potenza è graduale.
- **Tensione di Cut-Off (Low/Middle/High):** definisce la tensione di alimentazione dell’ESC (e quindi la tensione erogata dalla batteria) minima al di sotto della quale il motore non viene alimentato. Per batterie da 2 a 4 celle i valori sono rispettivamente 2.85V/3.15V/3.30V.
- **Modalità di avvio (Normal/Soft/Very soft):** in base al parametro impostato, la velocità del motore raggiungerà il valore massimo in un tempo più o meno breve. “Normal” è solitamente adatto agli aeroplani, mentre “Soft” e “Very Soft” vengono generalmente usati per i velivoli a decollo verticale.
- **Modalità di timing (Low/Middle/High)**
- **Numero di celle della batteria (Auto/2/3/4):** determina il numero di celle della batteria che si sta usando.

In Figura 14 è riportato un esempio di programmatore di ESC dove sono visualizzati i valori di default dei singoli parametri.



Figura 14: Scheda di programmazione di un'ESC

Per programmare l’ESC bisogna collegarlo alla scheda di programmazione utilizzando l’ingresso BEC o l’ingresso batteria. A questo punto è sufficiente utilizzare i pulsanti per modificare la configurazione dell’ESC rappresentata dai singoli led.

4.3.2 Procedura di calibrazione

Prima di utilizzare un’ESC è necessario eseguire una procedura detta di “Calibrazione”. Questa procedura è necessaria per impostare il range effettivo di variazione del duty cycle del segnale PWM che comanderà l’ESC, definendo quindi il minimo e il massimo valore del duty cycle che corrispondono rispettivamente alla minima e alla massima velocità del motore. I passaggi per la calibrazione sono i seguenti:

1. Si manda un segnale PWM avente il massimo duty cycle desiderato
2. Si connette la batteria all’ESC; l’ESC emetterà un breve segnale audio melodico e poi inizierà ad emettere un BEEP ogni 2 secondi;

3. Aspettare almeno due secondi;
4. Si manda un segnale PWM avente il minimo duty cycle desiderato
5. Aspettare qualche secondo finché l'ESC non emetterà rapidamente un numero di BEEP pari al numero di celle della batteria e a seguire un lungo BEEP per segnalare che la procedura di calibrazione è terminata correttamente.

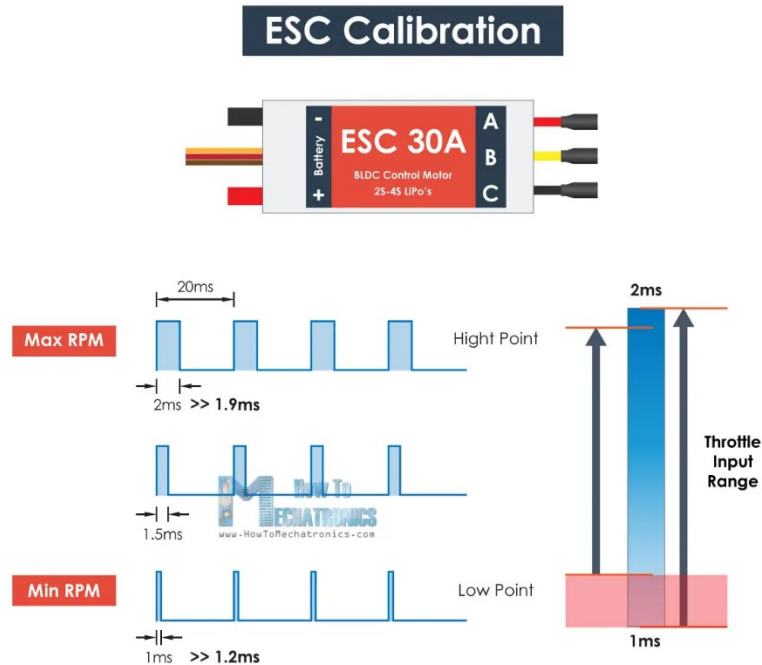


Figura 15: Effetto procedura di calibrazione

4.3.3 Procedura di armamento

L'armamento è una procedura di sicurezza che richiede di inviare all'ESC un PWM con uno specifico duty cycle per segnalare l'effettiva volontà di avviare i motori ed evitare quindi che essi partano per errore non appena viene collegata la batteria. Nello specifico, la procedura richiede di utilizzare un duty cycle leggermente inferiore a quello minimo impostato in fase di calibrazione.

4.4 Motore Brushless Turnigy D3536/9 910KV

Per questo progetto è stato usato un motore brushless Turnigy D3536/9 910KV.



Figura 16: Motore brushless Turnigy D3536/9 910KV

Le caratteristiche tecniche del motore utilizzato sono le seguenti:

KV	910kv
Fasi	3
Massima corrente	25.5A
Potenza massima	370W a 15V
Peso	102g
Batteria	2 ~ 4 Cell /7.4~ 14.8V
Diametro del pozzo	5mm
No corrente do carico	1.5A
Dimensione	35x36mm
Spinta max	1050g
Dimensione prop.	7.4V/12x5 14.8V/10x7
Resistenza interna	0.063 Ohm

I parametri che ci interessano nella scelta di un motore sono il valore di Kv, il massimo assorbimento e la potenza totale. Assorbimento e potenza sono relativi al voltaggio di alimentazione. Il Kv rappresenta il numero di giri in funzione ai volt di alimentazione. Ad esempio, un valore di Kv di 900 RPM/V significa che se lo alimentiamo con una cella LiPo da 3.7V il motore effettuerà 3330 rotazioni al minuto (900 x 3.7).

4.5 Eliche Slow Fly Electric Prop 9057SF 10x5

Al motore è stato collegato assialmente l'elica bipala Slow Fly Electric Prop 9057SF da 10x5 pollici capace di fornire una spinta, a piena velocità, di circa 0.632Kg, come indicato dal venditore. Queste eliche sono progettate specificamente per applicazioni elettriche e offrono una combinazione di efficienza e controllo. La loro dimensione e passo, 10x5, sono in armonia con il Motore Turnigy D3536/9 910KV, consentendo una spinta efficace e un volo stabile.

La spinta fornita è sufficiente a far volare senza problemi un velivolo di un peso inferiore a 0.632kg.



Figura 17: Eliche da 10x5 pollici

4.6 Encoder ATM10

L'encoder è un sensore che serve per misurare la posizione e/o variazione di posizione dell'oggetto sopra il quale è installato. In sostanza misurando la variazione di posizione possiamo arrivare a calcolare la velocità lineare o angolare dell'oggetto, nel mio caso la velocità angolare dell'albero motore. Esistono vari tipi di Encoder, essi si differenziano in base alla tecnologia utilizzata per effettuare le misurazioni (capacitivo, ottico, magnetico...) ed in base a ciò che effettivamente sono in grado di misurare (lineari, rotativi...). Nel mio caso l'Encoder ATM10 è modulare, incrementale, capacitivo ed in quadratura.



Figura 18: Visione 3D del Encoder ATM10 installato sul motore Brushless

È incrementale perché la misura della posizione angolare non è assoluta ma relativa, ovvero l'encoder fornisce un'indicazione incrementale riferita alla posizione precedente.

È in quadratura poiché ha come output due segnali ad onda quadra sfasati tra di loro di 90° gradi. Avere due fasi anziché una, comporta una maggiore risoluzione e la possibilità di capire se la rotazione avviene in senso orario o antiorario.

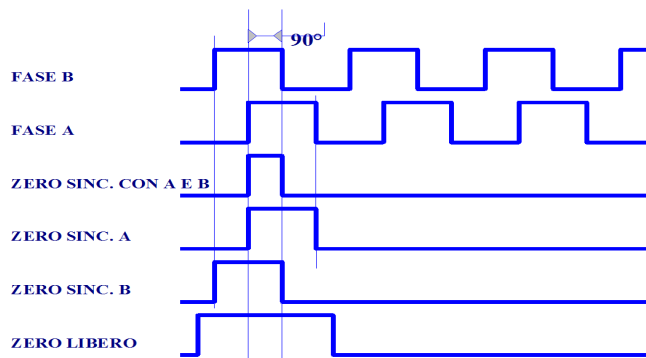


Figura 19: Fasi A e B dell'Encoder in quadratura

Secondo una convenzione riconosciuta da molti: l'incremento è positivo quando la fase B anticipa la fase A con rotazione dell'albero in senso orario, viceversa sarà antiorario se il canale A precede il canale B.

L'encoder è capacitivo poiché è composto da un rotore, un trasmettitore fisso ed un ricevitore fisso. Mentre ruota, il segnale di riferimento ad alta frequenza del trasmettitore viene modulato in modo prevedibile.

Il codificatore rileva le variazioni di capacità-reattanza sulla scheda del ricevitore e le traduce in incrementi di movimento rotatorio utilizzando un algoritmo di demodulazione.

1 = On, 0 = Off

Resolution (PPR)	Maximum RPM	1	2	3	4
2048	7500	0	0	0	0
1024	7500	0	0	1	0
1000	7500	1	0	0	0
800	7500	0	1	0	0
512	15000	0	0	0	1
500	7500	1	0	1	0
400	7500	0	1	1	0
384	7500	1	1	0	0
256	15000	0	0	1	1
250	15000	1	0	0	1
200	15000	0	1	0	1
192	7500	1	1	1	0
125	15000	1	0	1	1
100	15000	0	1	1	1
96	15000	1	1	0	1
48	15000	1	1	1	1

DIP switch:
Example setting: 500 PPR

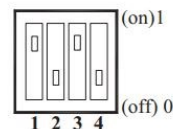


Figura 20: Settaggio del PPR del Encoder ATM-10

L'encoder è modulare poiché attraverso uno switch possiamo scegliere il valore di PPR (Pulse Per Revolution) desiderato.

Questo ci permette di avere una risoluzione variabile.

La precisione dell'encoder dipende, oltre che dalla risoluzione data dal PPR (più il PPR è elevato, tanto più è in grado di misurare piccole variazioni angolari), anche soprattutto dal K scelto.

Infatti, dalla teoria sappiamo che $\Delta\theta = \frac{360}{K * PPR}$; dove K assume valori 1,2 o 4. Se consideriamo un encoder a quadratura, come quello che stiamo analizzando, ciò significa che:

K = 1

Se contiamo solamente i fronti di salita o di discesa di un singolo canale.

K = 2

Se contiamo entrambi i fronti di salita e di discesa di un singolo canale.

K = 4

Se contiamo entrambi i fronti di salita e di discesa di entrambi i canali.

5 SOFTWARE

In questo capitolo vengono descritti brevemente gli strumenti software utilizzati in fasi di test. Nello specifico vengono presentati sia l'ambiente di sviluppo utilizzato per la programmazione della STM32 (STM32CubeIDE) sia lo strumento utilizzato per l'acquisizione dei dati tramite seriale (Serial Tools).

5.1 STM32CubeIDE

STM32CubeIDE è una piattaforma avanzata per lo sviluppo in C/C++ con funzionalità di configurazione delle periferiche, generazione del codice, compilazione del codice e strumenti di debug per microcontrollori e microprocessori STM32. Si basa sulla struttura Eclipse®/CDT™, sulla toolchain GCC per la compilazione e su GDB per il debugging. Consente l'integrazione di centinaia di plugin esistenti che completano le funzionalità dell'IDE Eclipse®.



STM32CubeIDE integra le funzionalità di configurazione delle periferiche e generazione automatica del codice da STM32CubeMX per offrire un'esperienza di sviluppo completa e risparmiare tempo di installazione e sviluppo. Dopo la selezione di un singolo microcontrollore/microprocessore STM32 o di una scheda che lo integra, viene creato un progetto vuoto e viene generato il codice di inizializzazione di default per il dispositivo selezionato. In qualsiasi momento durante lo sviluppo, l'utente può tornare nell'interfaccia grafica per la configurazione delle periferiche o del middleware e rigenerare il codice di inizializzazione senza influenzare il codice utente già sviluppato.

STM32CubeIDE include analizzatori di build e stack che forniscono all'utente informazioni utili sullo stato del progetto e sui requisiti di memoria. L'IDE include anche funzionalità standard e avanzate di debugging, tra cui visualizzazioni dei registri del core CPU, delle memorie e dei registri delle periferiche, nonché il monitoraggio delle variabili in tempo reale, l'interfaccia Serial Wire Viewer o l'analizzatore dei faults.

5.2 Serial Tools

Il monitoraggio e la visualizzazione in tempo reale di dati provenienti dalla STM32 durante i test sono essenziali per la diagnosi, l'ottimizzazione e il debug dell'applicazione sviluppata. In questo paragrafo, esploreremo l'uso di Serial Tools come alternativa a Putty su macOS per poter visualizzare cosa sta accadendo nel codice in esecuzione nella scheda. In fase di test questo strumento è ad esempio stato utilizzato per visualizzare il duty cycle del segnale PWM fornito all'ESC durante le prove (Figura 21).

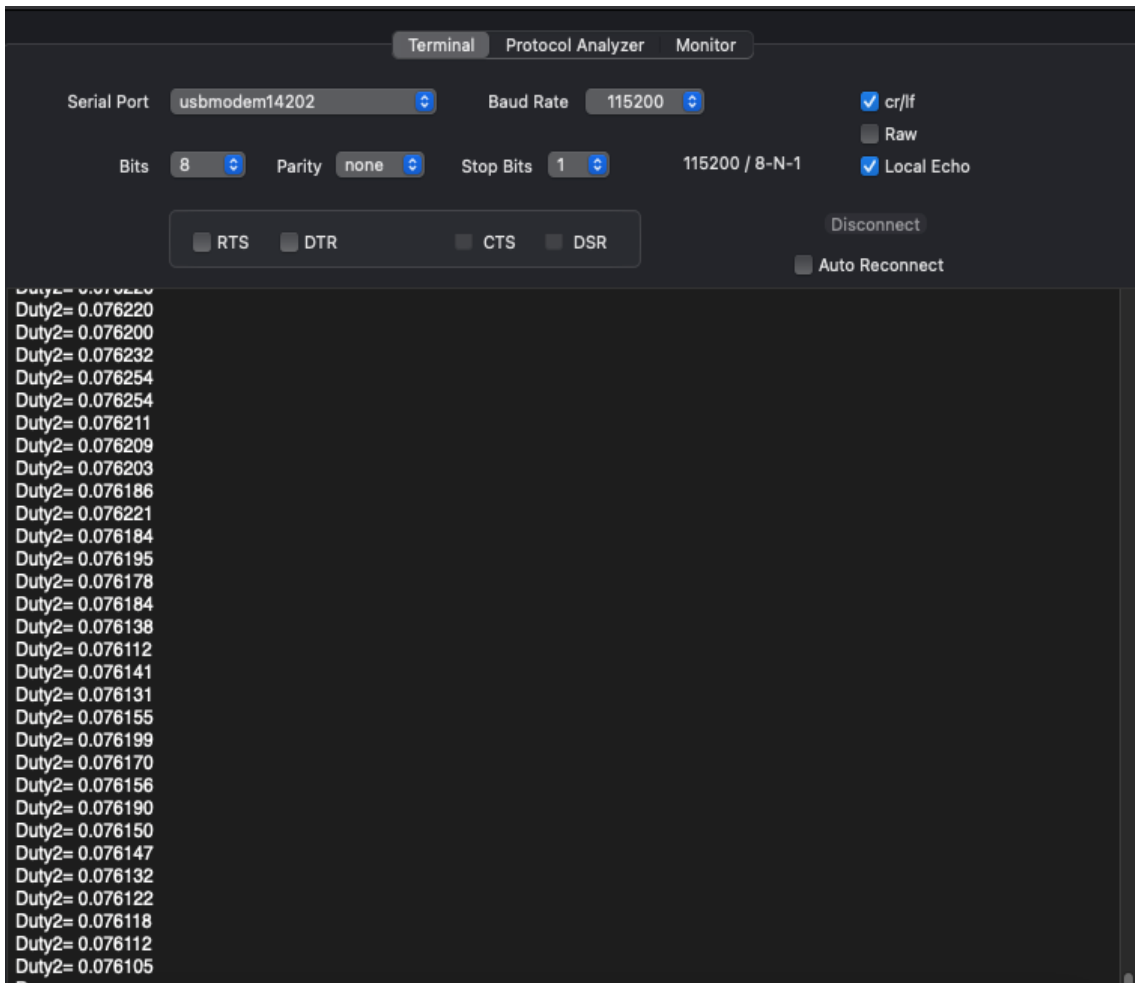


Figura 21: Serial Tools

Serial Tools è una piattaforma versatile che offre funzionalità del tutto analoghe a Putty, ma con un'interfaccia utente più user-friendly e adatta agli utenti di dispositivi iOS. Anche in questo caso per utilizzarla è sufficiente selezionare la porta seriale su cui è collegata l'STM32 e impostare correttamente i parametri della seriale che sono: baud rate, numero di bit dei dati, parità, numero di bit di stop. In Figura 21 si possono osservare i valori da impostare nel caso in cui sulla STM32 venga abilitata l'interfaccia USART collegata all'ST-Link mantenendone le impostazioni di default.

5.3 MATLAB

MATLAB è un ambiente di sviluppo e un linguaggio di programmazione utilizzato principalmente per l'analisi numerica, la visualizzazione di dati, la modellazione matematica e la simulazione. Il nome MATLAB è un'abbreviazione di "Matrix Laboratory", poiché la sua principale forza risiede nell'elaborazione efficiente di matrici e array multidimensionali.

Una delle caratteristiche distintive di MATLAB è la sua potente capacità di creare grafici e visualizzazioni. Questo è particolarmente utile per rappresentare dati sperimentali, risultati di simulazioni e altre informazioni visive.

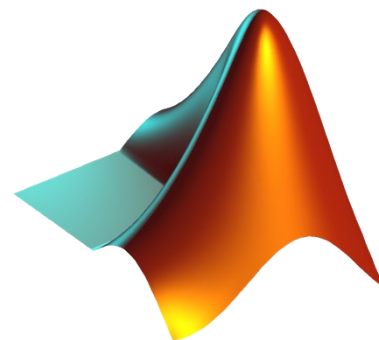


Figura 22: MATLAB

Nel corso del mio progetto, ho fatto utilizzo di MATLAB per esplorare e analizzare e la connessione tra il duty cycle e i giri del motore. MATLAB si è rivelato uno strumento cruciale nel fornire una chiara visualizzazione dell'andamento della curva del grafico, consentendomi di ottenere una comprensione approfondita della relazione dinamica tra questi due parametri fondamentali.

SVILUPPO DEL CODICE

In questo capitolo sono descritti e commentati tutti i principali codici implementati e utilizzati in fase di test, specificando sia come la scheda è stata configurata attraverso il file con estensione '.ioc', sia il codice che è stato sviluppato.

6 Configurazione periferiche

In fase di test sono state utilizzate diverse periferiche disponibili sulla STM32. Nello specifico le principali sono le seguenti:

- Timer in modalità PWM: è stato utilizzato per generare il segnale PWM che comanda l'ESC;
- ADC: è stato abilitato per poter leggere la posizione di un trimmer (o potenziometro); il trimmer è stato utilizzato per poter variare manualmente e in tempo reale il duty cycle del segnale PWM;
- USART/UART: è stata utilizzata per reindirizzare lo standard input/output in modo da poter visualizzare in tempo reale i dati inviati dalla scheda sul terminale del pc;

Nel seguito viene descritto come le singole periferiche sono state abilitate e configurate mediante l'interfaccia grafica offerta dall'STM32CubeMX integrato nel Cube IDE.

6.1 PWM

Per poter controllare il motore brushless attraverso l'ESC è necessario generare un segnale PWM con frequenza di 50Hz (periodo 20ms) e duty cycle variabile.

Sulle schede STM32 è possibile generare dei segnali PWM utilizzando i timer che supportano una modalità PWM dedicata. Il primo passo consiste nell'inizializzare e assegnare al core utilizzato (l'M4) un timer che supporta la modalità PWM. Per il progetto è stato scelto di utilizzare il Timer 3 e il suo Channel 3. A questo punto è necessario configurare in maniera opportuna i suoi parametri "Prescaler" e "Counter Period" in modo da generare un segnale PWM con frequenza desiderata.

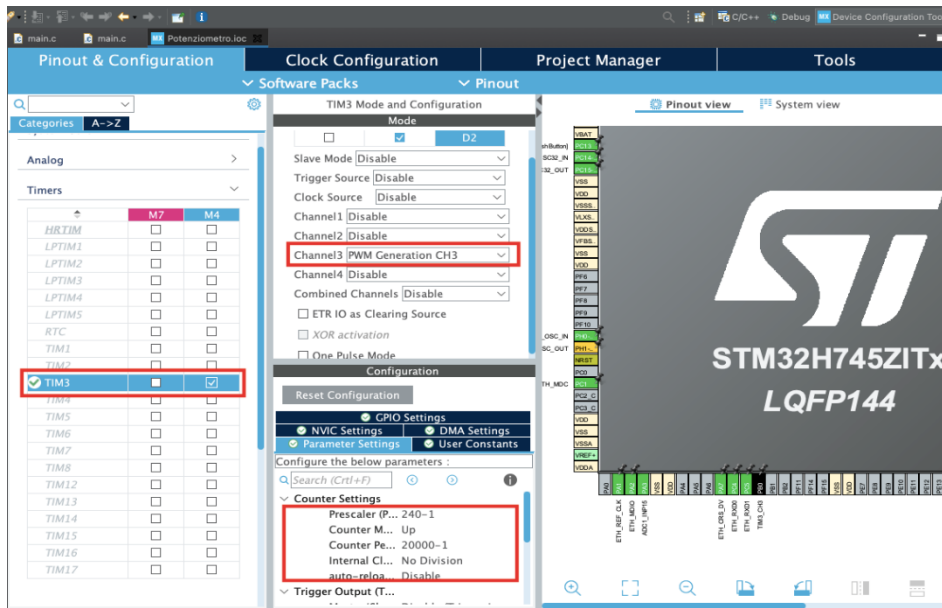


Figura 23: configurazione PWM

Il “Prescaler” è il divisore del clock di base del timer (Figura 24) e determina il clock effettivo che alimenta il timer e quindi la frequenza con cui il suo contatore viene incrementato. Il “Counter Period” determina invece il valore massimo del contatore raggiunto il quale il timer viene resettato. Combinando questi due parametri si determina la frequenza di reset del timer e quindi la frequenza del segnale PWM. Avendo un clock base del timer di 240 MHz (Figura 24) una delle possibili scelte dei parametri per ottenere una frequenza del PWM di 50Hz è la seguente:

- Prescaler = 240-1
- Counter Period = 20000-1

$$Timer_{freq} [s] = \frac{Prescaler}{Clock_{source}[Hz]} \cdot Period = \frac{240}{240'000'000} \cdot 20000 = 0,02 s = 20 ms$$

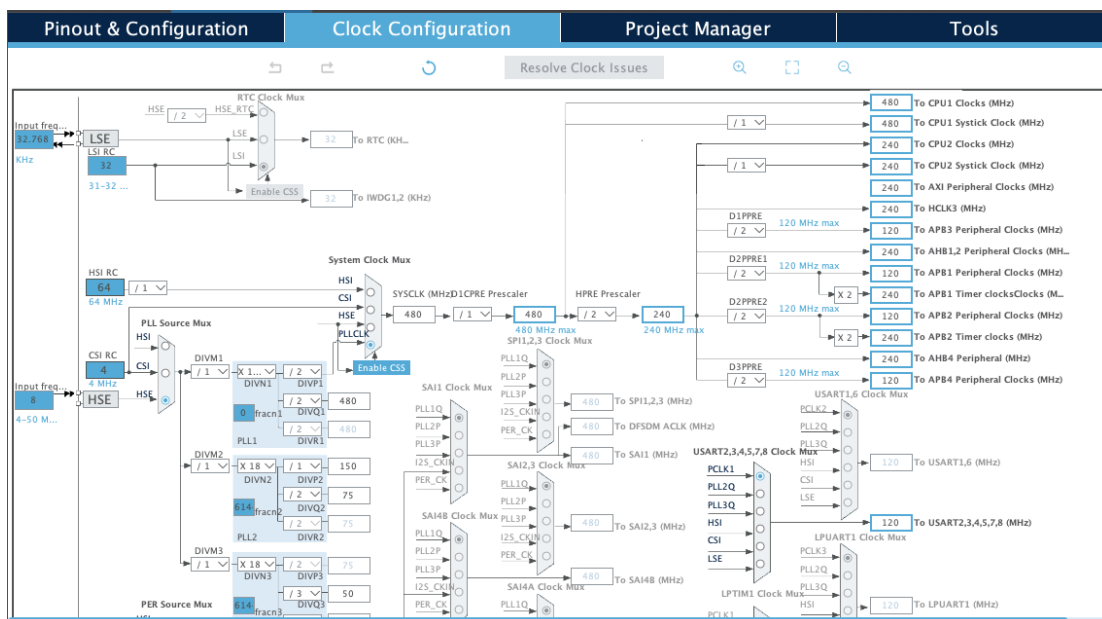


Figura 24: Clock configuration

Il duty cycle del segnale PWM verrà invece modificato a livello di codice in modo da controllare la potenza erogata al motore e quindi la sua velocità di rotazione. Per quanto

riguarda questo aspetto, nel file con estensione '.ioc' è stato necessario soltanto impostare il canale in "PWM mode 1" in modo che il PWM generato all'interno di ogni periodo parta con il segnale alto che viene poi abbassato quando il contatore del timer raggiunge il valore impostato nel registro CCR.

6.2 ADC

Il convertitore analogico-digitale (ADC) è un componente essenziale per tradurre segnali analogici in formati digitali, consentendone la lettura e l'elaborazione da parte di microcontrollori o processori. In questo progetto l'ADC è stato utilizzato per leggere la posizione di un trimmer.

L'STM32 dispone di più ADC ciascuno con diversi canali. Nel file con estensione '.ioc' è quindi necessario abilitare uno specifico ADC (e assegnarlo al core utilizzato) e specificare il canale che si intende utilizzare per la conversione. In questo progetto si è scelto di utilizzare il canale 15 dell'ADC1. L'ADC è inoltre stato utilizzato in modalità continua in modo da leggere in tempo reale la posizione del trimmer.

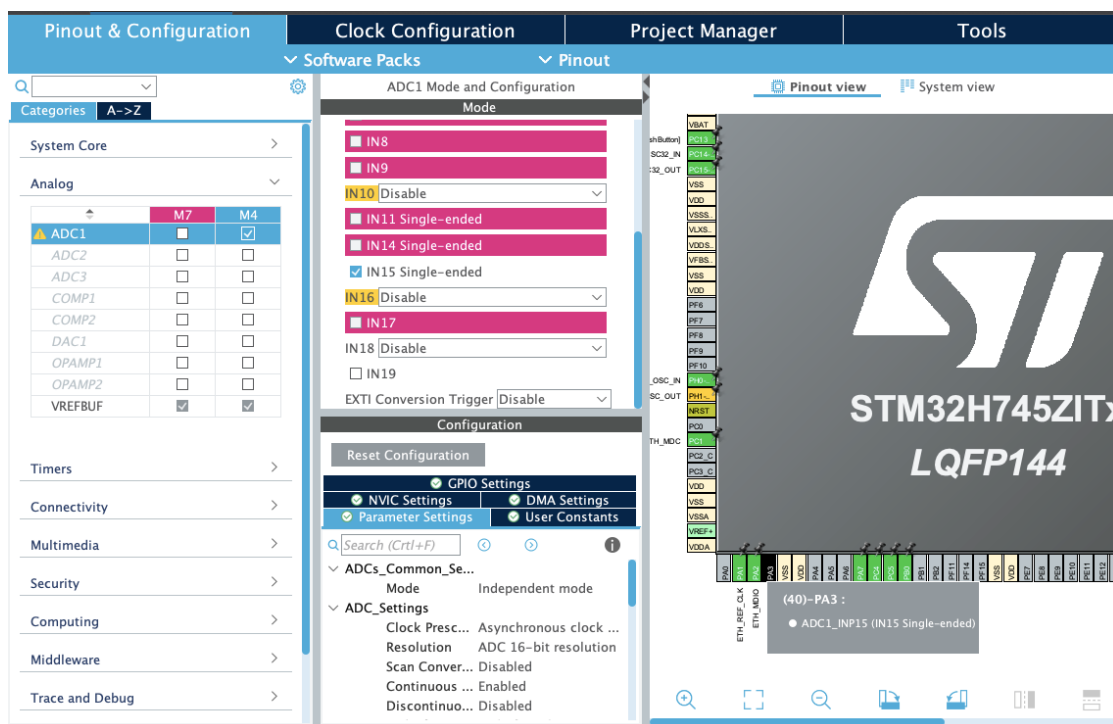


Figura 25: Configurazione ADC

6.3 Time Encoder mode

Utilizzando il file con estensione '.ioc', ho configurato il timer TIM2 in modalità encoder. La scelta di operare senza un prescaler è stata determinante per massimizzare la precisione delle letture dell'encoder. Questo approccio ha consentito al timer di conteggiare direttamente gli impulsi dell'encoder, evitando qualsiasi riduzione nella risoluzione delle letture.

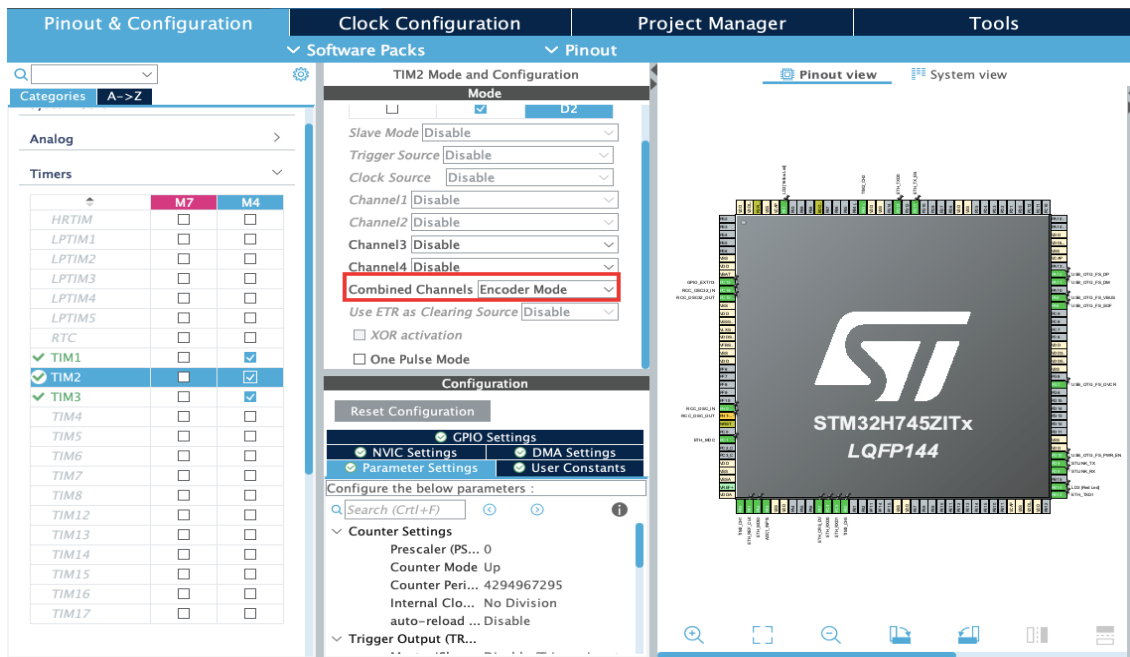


Figura 26: TIM2 Encoder mode

Per garantire una lettura continua e senza interruzioni, ho impostato il periodo massimo del timer TIM2. Questa precauzione è stata adottata per evitare il verificarsi di overflow, che potrebbe compromettere l'integrità delle letture dell'encoder. L'impostazione del periodo massimo ha assicurato una gestione ottimale dell'encoder, garantendo la continuità delle letture anche in condizioni di utilizzo prolungato.

7 Codice

Grazie all'STM32CubeMX integrato nell'IDE una volta configurato il microcontrollore in base alle proprie necessità utilizzando una semplice interfaccia grafica, è sufficiente richiedere la generazione automatica del codice di inizializzazione delle periferiche per tradurre in codice quanto richiesto. A questo punto è possibile modificare il codice generato per inserire la logica vera e propria. Questo capitolo è quindi dedicato alla spiegazione del codice che è stato implementato per effettuare i singoli test, mettendo in evidenza le scelte di progettazione e le soluzioni adottate.

7.1 Trimmer per la modifica del duty cycle

La funzione riportata di seguito rappresenta la callback chiamata quando una conversione ADC è completata per l'ADC1. Essa legge il valore convertito, calcola la tensione corrispondente, determina il rapporto ciclo di lavoro basato sulla tensione calcolata e aggiorna il registro di comparazione del canale 3 del timer TIM3. Questo processo è utile, ad esempio, per modificare dinamicamente il duty cycle di un segnale PWM in base ai valori letti da un sensore attraverso l'ADC.

In particolare, nella funzione proposta il duty utilizzato per generare il PWM ('duty1') corrisponde alla percentuale ricavata dall'elaborazione del segnale del trimmer ('duty0') applicata su un range [duty_min, duty_max] scelto in base al test da eseguire.

```

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc){
    if(hadc == &hadc1){
        n_count = HAL_ADC_GetValue(&hadc1);
        v = n_count * 3.3 / (65536-1);
        duty0 = v / 3.3; // v=3.3 -> duty = 1
        duty1 = duty_min + (duty_max - duty_min) * duty0;
        TIM3->CCR3 = duty1 * TIM3->ARR;
    }
}

```

7.2 Filtro utilizzato per il trimmer

In fase di test è emerso che la lettura diretta del trimmer attraverso l'ADC è particolarmente rumorosa: mantenendo fissa la posizione del trimmer la tensione letta dall'ADC, e quindi il duty cycle del PWM che pilota l'ESC, ha delle oscillazioni non accettabili.

Per cercare di risolvere questo problema il codice descritto nel paragrafo 7.1 è stato modificato aggiungendo un filtro, e nello specifico un semplice filtro a media mobile, con l'obiettivo di stabilizzare il valore del duty cycle prima di applicarlo al timer. Questo tipo di filtro è utilizzato per ridurre il rumore o le fluttuazioni in un segnale, calcolando la media di un certo numero di campioni consecutivi. Nel nostro caso è la funzione `Calc_Average` a calcolare la media di un array di valori di duty cycle (parametro 'media') la cui dimensione è specificata come parametro della funzione (parametro 'lunghezza'). Il valor medio così calcolato è restituito dalla funzione e costituisce il valore di duty cycle del segnale PWM effettivamente inviato all'ESC. Questo filtro può aiutare a rendere il valore del duty cycle più stabile nel tempo, riducendo il contributo di fluttuazioni o rumore nei singoli campioni. Di seguito la funzione `Calc_Average()` appena descritta:

```

float Calc_Average(float media[], int lunghezza){
    float somma =0.0;
    for(int i=0; i<lunghezza; i++)
        somma+=media[i];
    return somma/lunghezza;
}

```

Nel seguito si riporta invece tutta la logica del programma che è contenuta nella callback dell'ADC, richiamata ogni volta che una nuova misura è disponibile (l'ADC è infatti stato configurato in modalità continua e in interrupt).

```

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc){
    if(hadc == &hadc1){
        n_count = HAL_ADC_GetValue(&hadc1);
        v = n_count * 3.3 / (65536-1);
        duty0 = v / 3.3; // v=3.3 ->duty = 1
        duty1=duty_min + (duty_max-duty_min)* duty0;
        filtro[P]=duty1;
        P=P +1;
        if (P>=size)
            P=0;
        duty2= Calc_Average(filtro,size);
        TIM3->CCR3 = duty2*TIM3->ARR;
    }
}

```

Il codice si limita ad acquisire l'uscita dell'ADC, trasformarla in duty cycle, filtrarla con la funzione Calc_Average () precedentemente descritta ed infine a modificare il duty cycle del segnale PWM impostandolo al valore filtrato.

Per fare questo è stato utilizzato un array circolare (variabile 'filtro') di una certa lunghezza (variabile 'size'). Ogni volta che un nuovo campione viene acquisito, viene aggiunto all'array e l'indice (variabile 'P') viene aggiornato. Se l'indice assume come valore la lunghezza massima dell'array viene riportato a 0 per sovrascrivere i campioni più vecchi. Viene quindi richiamata la funzione Calc_Average che calcola la media degli elementi nell'array 'filtro', fornendo una versione "filtrata" del duty cycle (variabile 'duty2'), che è meno sensibile al rumore del trimmer. Infine, un segnale PWM col duty cycle appena calcolato viene effettivamente generato modificando il registro CCR del timer.

7.3 Calibrazione dell'ESC

La calibrazione è una procedura che consiste nel definire il minimo e il massimo valore di duty cycle del segnale PWM che riceverà l'ESC e che quest'ultima farà corrispondere rispettivamente alla minima e alla massima velocità del motore. I passaggi per la calibrazione che il codice deve implementare sono i seguenti:

1. Mandare un segnale PWM avente il massimo duty cycle desiderato (ad esempio 10%) prima che la batteria sia collegata;
2. Mandare un segnale PWM avente il minimo duty cycle desiderato (ad esempio 5%) una volta che il massimo duty è stato correttamente acquisito.

Per poter fare questo è stato utilizzato il pulsante blu (User Button) in modo da poter variare il duty cycle dal minimo al massimo con la sua pressione.

Sono state introdotte due variabili globali min_duty e max_duty che devono essere inizializzate rispettivamente al valore minimo e massimo del duty cycle desiderati. Il codice che imposta inizialmente il duty cycle al valore massimo è il seguente:

```

HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_3);
TIM3->CCR3 = (max_duty/100)*TIM3->ARR;

```

Il codice si limita ad avviare il timer in modalità PWM e ad impostarne il duty cycle assegnando al registro CCR del timer il valore opportuno, cioè la percentuale del registro ARR (Auto-Reload Register contenente il periodo del timer) corrispondente al duty cycle desiderato.

La callback del pulsante, che modifica il duty al valore minimo, è invece la seguente:

```
void HAL_GPIO_EXIT_Callback(uint16_t GPIO_Pin){
    if(GPIO_Pin==GPIO_PIN_13){
        TIM3->CCR3 = (min_duty/100)*TIM3->ARR;
    }
}
```

7.4 Armatura dei motori

Come già spiegato in precedenza, prima di poter controllare in velocità un motore brushless questo deve essere armato. L'armamento è una procedura di sicurezza che richiede di mandare un segnale PWM con duty cycle minimo (il duty cycle minimo impostato in fase di calibrazione). Ipotizzando quindi ad esempio di aver impostato in fase di calibrazione un duty cycle minimo del 4.75% con una frequenza di 50Hz, è questo il segnale PWM che deve essere fornito all'ESC per armare il motore. Di seguito il codice che consente di generare il segnale PWM e di impostarne il duty cycle al valore opportuno:

```
min_duty = 4.75;
HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_3);
TIM3->CCR3 = (min_duty/100)*TIM3->ARR;
HAL_Delay(2000);
```

Se la fase di armamento si conclude con successo l'ESC emetterà un numero di BEEP corti veloci pari al numero di celle della batteria e a seguire un BEEP lungo (analogamente a quanto avviene al termine della procedura di calibrazione). A questo punto è possibile controllare la velocità del motore variando il duty cycle all'interno del range impostato in fase di calibrazione. Ad esempio, è possibile avere un range di controllo della velocità è compreso fra 5% (velocità minima) e 15% (velocità massima) di duty cycle. Questi valori si possono però appunto cambiare con la procedura di calibrazione dell'ESC.

7.5 Codice complessivo per test del motore

Prima di eseguire il codice di test principale, è essenziale calibrare il sistema. La calibrazione coinvolge l'utilizzo di specifici comandi per impostare i parametri iniziali. Una volta completata la calibrazione, è possibile eseguire il codice di test che include operazioni come l'utilizzo di armatura e trimmer con filtro. Di seguito è riportato il codice utilizzato nella funzione main e le funzioni implementate:

```

HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_3);
TIM3->CCR3 = (max_duty/100)*TIM3->ARR;

min_duty = 4.75;
HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_3);
TIM3->CCR3 = (min_duty/100)*TIM3->ARR;
HAL_Delay(2000);

float Calc_Average(float media[], int lunghezza){
    float somma =0.0;
    for(int i=0; i<lunghezza; i++)
        somma+=media[i];
    return somma/lunghezza;
}

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc){
    if(hadc == &hadc1){
        n_count = HAL_ADC_GetValue(&hadc1);
        v = n_count * 3.3 / (65536-1);
        duty0 = v / 3.3; // v=3.3 ->duty = 1
        duty1=duty_min + (duty_max-duty_min)* duty0;
        filtro[P]=duty1;
        P=P +1;
        if (P>=size)
            P=0;
        duty2= Calc_Average(filtro,size);
        TIM3->CCR3 = duty2*TIM3->ARR;
    }
}

void HAL_GPIO_EXIT_Callback(uint16_t GPIO_Pin){
    if(GPIO_Pin==GPIO_PIN_13){
        tIM3->CCR3 = (min_duty/100)*TIM3->ARR;
    }
}

```

7.6 Codice Lettura Encoder

In questa parte del codice, ci concentriamo sulla fase cruciale di elaborazione dei dati provenienti da un encoder per ottenere informazioni significative sulla velocità angolare di un sistema. Il codice seguente è parte integrante di un'applicazione che sfrutta il timer TIM2 su un microcontrollore STM32 per la gestione degli impulsi di un encoder. Vediamo nel dettaglio come questi dati vengono elaborati per ottenere la velocità angolare.

```

diff_count_encoder = count_encoder - ref_count_encoder;
delta_angle_deg = ((float)diff_count_encoder) * 360.0 /
((float)ENCODER_COUNTING_MODE*ENCODER_PPR);
speed = delta_angle_deg/dt;
speed_RPM = speed *60/360;

```

Questo blocco di codice fornisce una chiara comprensione di come i dati dell'encoder vengono trasformati in informazioni utili sulla velocità angolare. I risultati ottenuti, come la velocità in RPM, possono essere utilizzati per monitorare e regolare dinamicamente il comportamento di un sistema motorizzato.

7.7 Codice con incremento dinamico del duty cycle

Questo frammento di codice è parte di un sistema di controllo dinamico che gestisce il ciclo di lavoro di un motore utilizzando il Duty Cycle ottenuto dal segnale PWM (Pulse Width Modulation) e raccoglie informazioni cruciali sulla velocità angolare attraverso un encoder.

```
if(flag_disarm != 1){
    if(flag_Tc == 1){
        flag_Tc = 0;

        inc = inc + 1;
        if((inc%5) == 0){
            duty_inc = duty_inc + 0.0005;
        }
        TIM3->CCR3 = duty_inc*TIM3->ARR;

        diff_count_encoder = count_encoder - ref_count_encoder;
        delta_angle_deg = ((float)diff_count_encoder) * 360.0 /
        ((float)ENCODER_COUNTING_MODE*ENCODER_PPR);
        speed = delta_angle_deg/dt;
        speed_RPM = speed *60/360;
        printf("Duty = %f, Speed RPM: %f\r\n",duty_inc, speed_RPM);
    }
} else {
    printf("Disarm duty min \r\n");
    HAL_Delay(100);
}
```

L'incremento periodico del ciclo di lavoro fornisce una modalità dinamica di adattamento del controllo del motore. La velocità angolare, derivata dall'encoder, viene calcolata e stampata sulla console, fornendo un feedback in tempo reale sulle prestazioni del sistema.

L'incremento del ciclo di lavoro `duty_inc` è progettato per essere incrementato ogni 5 iterazioni del loop principale. Questo può essere interpretato come un modo per variare dinamicamente il ciclo di lavoro del PWM nel tempo.

Il valore specifico 0.0005 è aggiunto a `duty_inc` ad ogni passo, il che significa che il ciclo di lavoro del segnale PWM aumenta gradualmente. Questo tipo di approccio può essere utile per ottenere una variazione graduale della potenza erogata al motore, regolando così la velocità del motore in modo più controllato.

7.8 Codice Matlab

Nel corso delle sperimentazioni e delle analisi condotte sulla piattaforma STM, è emersa la necessità di implementare una metodologia efficace per acquisire e visualizzare i dati dei test in modo chiaro e comprensibile. Per raggiungere questo obiettivo, è stato sviluppato un codice Matlab dedicato, il quale svolge un ruolo fondamentale nel processo di analisi e interpretazione dei risultati ottenuti.

Il codice è progettato per interagire con la STM attraverso la comunicazione USART, consentendo l'acquisizione dei dati direttamente dalla piattaforma.

Il codice genera grafici chiari e descrittivi che rappresentano in modo visuale le relazioni dinamiche tra il duty cycle e la velocità del motore. Questi grafici sono riportati nel capitolo dei Test e Grafici (9), essenziali per una comprensione approfondita dei comportamenti del sistema.

```
clearvars;
close all;
clc;

%% Parametri acquisizione
port = 'COM3';
Baud_Rate = 115200;
Data_Bits = 8;
Stop_Bits = 1;

if ~isempty ( instrfind )
    fclose ( instrfind );
    delete ( instrfind );
end
s = serial (port , 'BaudRate', Baud_Rate , 'DataBits' , Data_Bits , 'StopBits',
Stop_Bits );

% Aspettare che la porta sia disponibile
while ( isempty ( seriallist ) )
end
fopen (s);

%% Acquisizione
i=1;
while true
    try
        A = fscanf (s, "Duty = %f, Speed RPM: %f\r\n");
    catch
        fscanf (s, "%f");
        A=[];
        break
    end
    if isnumeric(A) && all(size(A)~=0)
        duty(i)=A(1);
        speed(i)=A(2);
        fprintf("Acquisito: %i, %i\n",duty(i),speed(i));
        i=i+1;
    else
        fprintf("Non acquisito: %s\n",A);
    end
end

%% Chiusura porta
fclose (s);
delete (s);
clear s;
```

```

%% Elaborazione dati
ValDuty=unique(duty);

for i=1:length(ValDuty)
    j=find(duty==ValDuty(i));
    duty(j);
    speed(j);
    l=2;
    if length(j)<5
        if length(j)<3
            l=length(j)-1;
        end
    end
    ValSpeed(i)=mean(speed(j(end-l:end)));
end

%% Dati batteria
InitialBatteryVoltage = 16.4;
FinalBatteryVoltage   = 15.8;
CalibrationRange      = "4% -14%";
id=3;

%% Rappresentazione grafica
f1=figure('units','normalized','outerposition',[0 0 1 1]);
set(f1,'color','w');
plot(duty*100,speed,'*k',ValDuty*100,ValSpeed,'or');
hold on;
yline(InitialBatteryVoltage*910,'r');
yline(FinalBatteryVoltage*910,'r');
grid minor;
text(ValDuty(end)*100,InitialBatteryVoltage*910+250,sprintf("%2.1f V",InitialBat-
tteryVoltage));
text(ValDuty(end)*100, FinalBatteryVoltage*910-250,sprintf("%2.1f V",FinalBat-
tteryVoltage));
xlabel("duty (%)");
ylabel("speed (RPM)");
saveas(f1,sprintf("PuntiAcquisiti%i.jpg",id))

f2=figure('units','normalized','outerposition',[0 0 1 1]);
set(f2,'color','w');
plot(ValDuty*100,ValSpeed,'-k');
hold on;
yline(InitialBatteryVoltage*910,'r');
yline(FinalBatteryVoltage*910,'r');
grid minor;
xlabel("duty (%)");
ylabel("speed (RPM)");
text(ValDuty(end)*100,InitialBatteryVoltage*910+250,sprintf("%2.1f V",InitialBat-
tteryVoltage));
text(ValDuty(end)*100, FinalBatteryVoltage*910-250,sprintf("%2.1f V",FinalBat-
tteryVoltage));
saveas(f2,sprintf("Curva%i.jpg",id))

```

```

%% Scrittura file
fileID = fopen(sprintf("Report%i.txt",id), 'w');
fprintf(fileID, 'Date: %s\n', datestr(now, 'HH:MM:SS.FFF'));
fprintf(fileID, "Battery voltage at the beginning: %2.1f\n", InitialBatteryVoltage);
fprintf(fileID, "Battery voltage at the ending: %2.1f\n", FinalBatteryVoltage);
fprintf(fileID, "Calibration range: %s\n", CalibrationRange);
fprintf(fileID, "%s\nFULL ACQUISITION DATA\n%s\nDUTY(x/100) SPEED(RPM)\n", repmat('-',
1,60), repmat('-',1,60));
for i=1:length(duty)
    fprintf(fileID, "    %1.4f;      %6.2f;\n", duty(i), speed(i));
end
fprintf(fileID, "\n%s\nELABORATED DATA\n%s\nDUTY(x/100) SPEED(RPM)\n", repmat('-',
1,60), repmat('-',1,60));
for i=1:length(ValDuty)
    fprintf(fileID, "    %1.4f;      %6.2f;\n", ValDuty(i), ValSpeed(i));
end
fclose(fileID);

%% Salvataggio workspace
save(sprintf("Workspace%i.mat",id));

```

STUDIO FINALE

Il codice descritto nel capitolo precedente è stato usato per eseguire una serie di test che avevano come obiettivo principale quello di comprendere l'effetto della procedura di calibrazione e dei parametri dell'ESC sull'armatura e soprattutto sul controllo della velocità del motore.

In particolare, si è cercato di rispondere alle seguenti domande:

1. Qual è la relazione tra il duty cycle per l'armamento, il duty cycle minimo impostato in fase di calibrazione e il minimo duty cycle da fornire affinché il motore inizi a ruotare?
2. Ci sono dei limiti nella scelta del range del duty cycle che viene fatta con la procedura di calibrazione?
3. Che effetto hanno i parametri dell'ESC?

Tutti i test sono stati effettuati modificando i parametri dell'ESC dal programmatore e/o variando il range di duty cycle in fase di calibrazione per poi testare e comprendere il comportamento effettivo del motore al variare del duty cycle del segnale PWM usando il trimmer.

8 Ottimizzazione del Controllo del Motore Brushless

Nel corso di questa ricerca dedicata all'ottimizzazione del controllo del motore brushless, sono stati sviluppati due setup distinti, ciascuno mirato a esplorare e perfezionare specifici aspetti del sistema. Questi due approcci forniranno una panoramica esaustiva delle potenzialità di controllo e regolazione del motore, evidenziando le loro peculiarità e implicazioni pratiche.

• Primo Setup 1

Il primo setup si concentra sulla fase fondamentale di calibrazione dell'ESC, l'armatura del motore brushless e la stabilità operativa. Utilizzando un potenziometro, il duty cycle viene variato dinamicamente, consentendo una regolazione in tempo reale della potenza erogata al motore. Questo approccio fornisce una visione dettagliata delle dinamiche di calibrazione e delle risposte del motore a variazioni immediate del duty cycle.

L'obiettivo è comprendere come la calibrazione influisca sul comportamento del motore e come la sua stabilità operativa si mantenga attraverso la variazione del duty cycle.



Figura 27: Setup1

Questo setup è dedicato al controllo e alla stabilizzazione di un motore brushless all'interno di un asse di un drone. Il sistema è composto da componenti chiave, tra cui la batteria, l'ESC, il motore brushless e l'elica, come precedentemente discusso. L'obiettivo

principale è creare un ambiente di test in cui il motore può essere controllato e stabilizzato in modo efficiente per garantire una risposta precisa e stabile alle variazioni del duty cycle.

- **Secondo Setup 2**

Il secondo setup introduce un approccio più dinamico, concentrandosi sull'incremento progressivo del duty cycle e sfruttando il feedback di un encoder per monitorare e associare i giri del motore a specifici valori di duty cycle. Questo setup non solo differisce nella modalità di calibrazione rispetto al primo, ma si basa anche sull'utilizzo dell'encoder per acquisire informazioni precise sulla velocità e la posizione del motore. L'obiettivo è esplorare come questa implementazione influisca sulla precisione del controllo e sulla correlazione tra il duty cycle e le prestazioni effettive del motore.

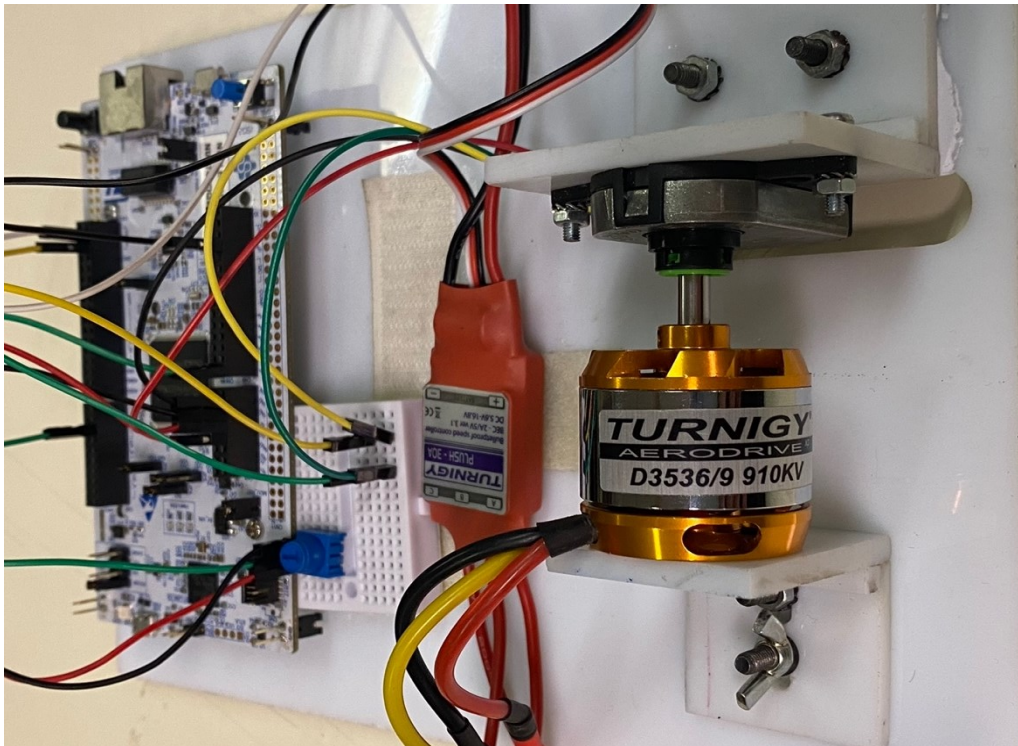


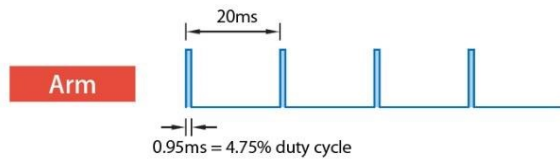
Figura 28:Setup2

Questo setup è dedicato all'analisi della relazione dinamica tra il duty cycle e i giri del motore brushless attraverso l'utilizzo di un encoder. Il sistema, composto dai componenti chiave quali la batteria, l'ESC, il motore brushless e l'encoder, è stato configurato per fornire una comprensione approfondita di come la variazione del duty cycle influisca sulla velocità di rotazione del motore.

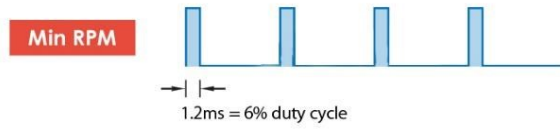
8.1 Calibrazione e Armamento (setup 1)

Dai test effettuati è emerso che, impostando in fase di calibrazione un range di duty cycle del [5%-10%], si ottiene quanto illustrato in Figura 15.

$$Arm = \frac{20 \times 4.75}{100} = 0.95ms$$



$$Min = \frac{20 \times 6}{100} = 1.2ms$$



$$Max = \frac{20 \times 10}{100} = 2ms$$

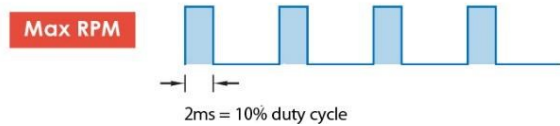


Figura 29: Effetto della calibrazione

8.2 Range di duty cycle per la calibrazione (setup1)

La scelta di un intervallo di duty cycle compreso tra il 5% e il 10% è comune quando si regola un ESC (Electronic Speed Controller) per motori brushless. Questa impostazione, ereditata dal controllo dei servomotori, rappresenta tipicamente il range di funzionamento standard per molti motori e applicazioni.

In questo studio abbiamo quindi cercato di capire quanto è possibile discostarsi da questo range standard. Ad esempio, calibrando l'ESC in un range [5%-60%] si è osservato quanto segue:

Prima dell'applicazione del Duty Cycle minimo (0-5%):

Arresto del Motore: finché il duty cycle non raggiunge il valore minimo impostato al 5%, il motore è completamente spento.

Nell'intorno del Duty Cycle minimo (5-7.6%):

Continuando ad aumentare progressivamente il duty cycle, si è osservato che il motore rimane spento finché non viene raggiunto un duty cycle soglia di circa il 7.6% (Figura 30) in corrispondenza del quale il motore inizia a ruotare alla velocità minima.

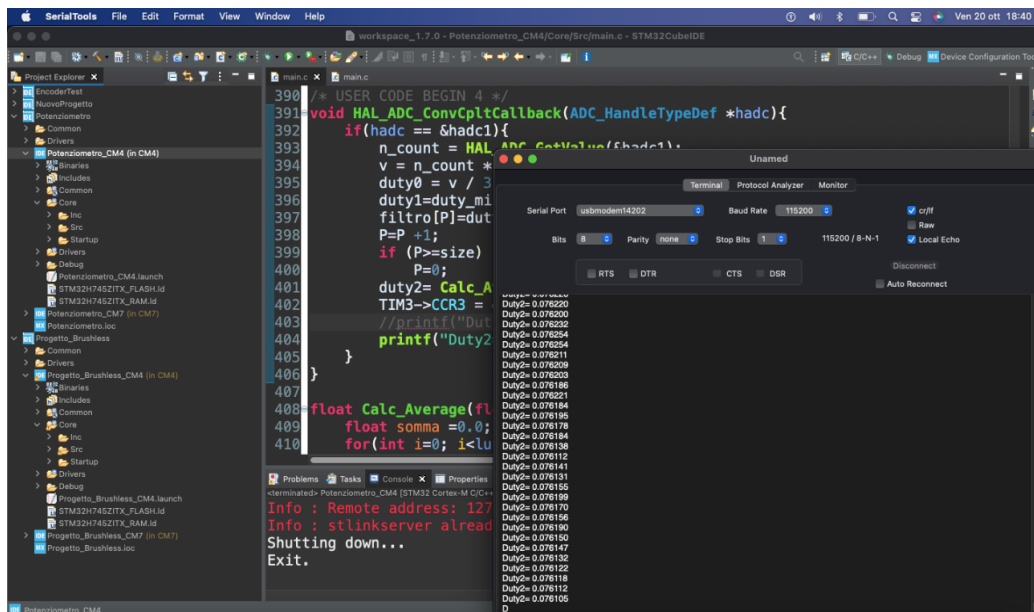


Figura 30: Test duty minimo

Applicando il Duty Cycle nella restante parte del range impostato in calibrazione (7.6-60%):

Avvio del Motore: aumentando gradualmente il duty cycle dal 7.6% al 60%, il motore inizia a ruotare. La velocità del motore aumenta man mano che il duty cycle cresce ma soltanto finché il duty cycle non raggiunge una soglia pari a circa il 14.7% (Figura 31). L'intervallo tra il 7.6% e il 14.7% rappresenta quindi l'unico range in cui si ha un controllo preciso sulla velocità di avvio e di stabilità della velocità, contrariamente a quanto atteso dal momento che in fase di calibrazione era stato impostato un range molto più ampio [5-60%]. Quando il duty cycle raggiunge e supera la soglia del 14.7% il motore si spegne.

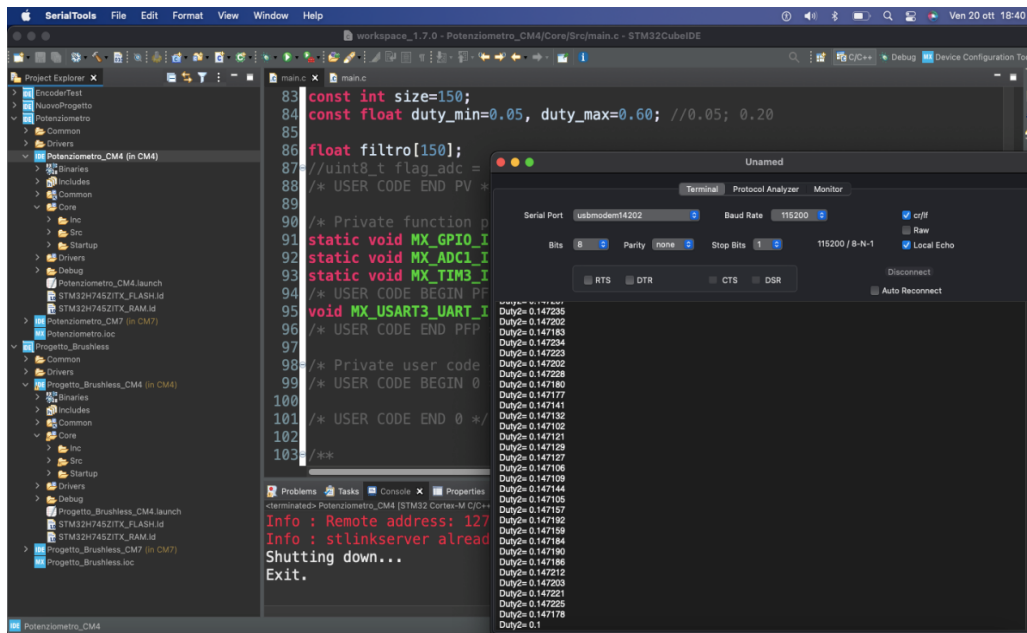


Figura 31: Test duty massimo

Alla luce di quanto osservato, si è cercato di comprendere il motivo di questo comportamento inatteso.

Una possibile giustificazione è che il motore si spegne al di sopra di una certa soglia perché entra in uno stato di protezione. Questo meccanismo di protezione è implementato per evitare il surriscaldamento del motore e prevenire eventuali danni causati da un funzionamento prolungato a livelli estremi di duty cycle. Un duty cycle più elevato richiede più energia per mantenere il motore a una velocità più alta. Di conseguenza, il consumo di energia aumenterà all'aumentare del duty cycle.

Un'altra possibile giustificazione è che sia il firmware dell'ESC a non permettere dei range di duty cycle che si discostano troppo da quello standard, nonostante la procedura di calibrazione sembri andare a buon fine.

In sintesi, per non incorrere in questi problemi, si consiglia una scelta dell'intervallo di duty cycle nell'intorno del range standard del [5%-10%] in modo da poter regolare con precisione la velocità in tutto il range impostato in fase di calibrazione. Questa impostazione offre un buon equilibrio tra controllo e prestazioni, consentendo di adattare il motore alle esigenze specifiche dell'applicazione. Al di fuori di questo intervallo, c'è il rischio di perdere il controllo del motore, soprattutto nella parte alta del range impostato, rendendo quindi inutile la procedura di calibrazione.

8.3 Effetto della modalità di avvio dell'ESC

Tra i parametri dell'ESC che possono essere impostati utilizzando il programmatore dedicato, come mostrato nel paragrafo (4.3.1), c'è la modalità di avvio. La modalità di avvio dell'ESC permette di configurare la velocità con cui l'ESC aumenta il "duty cycle" della tensione fornita alle fasi del motore e quindi la velocità con cui il motore risponde al comando di accelerazione. Tale parametro svolge un ruolo cruciale nell'influenzare la risposta e le prestazioni complessive del motore.

Ci sono tre modalità di avvio comuni nell'ESC: "Normale", "Soft" e "Super-Soft", come descritto nel manuale dell'Esc [\[1\]](#), ognuna caratterizzata da un tempo di accelerazione specifico:

Modalità di Avvio "Normale" (300 ms):

In questa modalità, l'ESC aumenta il "duty cycle" del segnale del motore in modo relativamente rapido, accelerando il motore in modo tempestivo. Ciò significa che il motore raggiungerà velocità elevate in un breve lasso di tempo. Questa modalità è preferita per aeromodelli e veicoli in cui è richiesta una risposta rapida del motore.

Modalità di Avvio "Soft" (6 secondi):

Questa modalità implica un aumento graduale del "duty cycle" quando viene richiesto un aumento della velocità del motore. Il motore aumenta quindi la sua velocità in modo più lento, il che è utile in situazioni in cui è necessaria una risposta del motore più delicata, come negli elicotteri.

Modalità di Avvio "Super-Soft" (12 secondi):

Questa modalità è simile alla modalità "Soft" ma con un tempo di accelerazione ancora più lungo. Il motore impiega più tempo per raggiungere la velocità massima. È ideale per applicazioni in cui il controllo preciso è essenziale, come in elicotteri e droni specializzati.

In sintesi, la modalità di avvio svolge un ruolo chiave nella regolazione del "duty cycle" della tensione di alimentazione delle fasi del motore generata dall'ESC, determinando quanto velocemente il motore raggiunge la velocità massima. La scelta della modalità appropriata è fondamentale per ottenere le prestazioni desiderate nell'applicazione.

8.4 Replicazione dei test su setup alternativo (setup2)

Nel corso di questa fase della ricerca, l'attenzione è stata rivolta ai test eseguiti su un secondo setup (Setup2), utilizzando gli stessi componenti, come la batteria, l'ESC, il motore brushless e l'encoder descritte nel capitolo (4), e seguendo le stesse procedure descritti nel paragrafo (8.2) e riportato nella figura (Figura 28). Questa sezione è finalizzata a comprendere come il secondo setup risponda alle stesse procedure di calibrazione e agli stessi test di variazione del duty cycle, esplorando eventuali differenze riscontrate.

Durante la procedura di calibrazione, si è notato che il secondo setup presenta una particolarità interessante rispetto al primo. La calibrazione ha avuto successo solo quando il duty cycle minimo è stato impostato all'interno del range del 4-5%, mentre non sono stati riscontrati limiti per il duty cycle massimo. Tuttavia, impostando valori di

duty cycle troppo bassi o troppo elevati ad esempio il 3% e il 18% si è verificato un ingresso in modalità di programmazione, come descritto nel manuale dell'ESC [1].

Particolarmente rilevante è stato il comportamento anomalo riscontrato impostando il duty cycle tra il 5% e il 10%. In questo intervallo, il sistema è entrato in uno stato di allarme o protezione, emettendo una sequenza di segnali acustici continui e frequenti non descritta nel manuale dell'ESC. Questo comportamento indica la presenza di limiti operativi o configurazioni critiche che richiedono ulteriori indagini e analisi.

I risultati preliminari suggeriscono che il secondo setup potrebbe reagire in modo diverso rispetto al primo, con una sensibilità particolare agli estremi del duty cycle. Questa variabilità potrebbe derivare da differenze nella configurazione interna dell'ESC o da parametri specifici del motore brushless.

Per migliorare la comprensione di tali comportamenti, si suggerisce che le fasi successive della ricerca si concentreranno sull'analisi approfondita delle specifiche tecniche dell'ESC e del motore brushless in uso.

8.5 Misura della velocità tramite encoder

Prima di procedere con l'analisi della relazione tra il duty cycle e la velocità del motore nel secondo setup, è stato fondamentale eseguire una serie di test dedicati a verificare l'affidabilità e la precisione dell'encoder nel monitorare la velocità di rotazione del motore brushless.

I test iniziali hanno evidenziato un'anomalia significativa nella misurazione della velocità del motore, come chiaramente documentato nella (Figura 32). I dati raccolti mostravano valori di velocità instabili, con variazioni anche nel segno. Inizialmente, è stata formulata l'ipotesi che questa problematica potesse essere correlata alla natura discontinua del movimento del motore brushless, che opera attraverso movimenti a scatti invece di un movimento continuo e fluido.

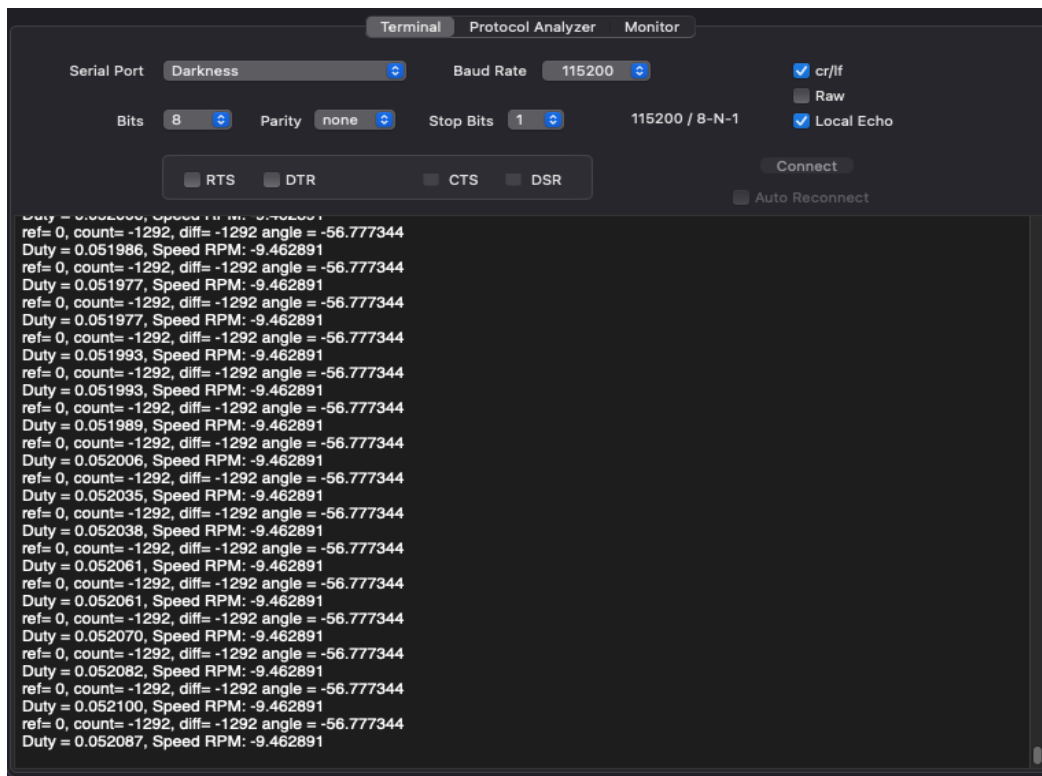


Figura 32: Verifica errori dell'encoder

Un'analisi più approfondita della problematica ha permesso di individuare che il malfunzionamento non era da attribuire al tipo di motore, bensì all'encoder stesso. Lo smontaggio dell'encoder ha rivelato che due componenti interni del sensore si erano staccati come si evidenzia nella Figura 33, dove si evidenziano la parte nera, che si collega all'albero motore, e il disco verde all'interno dell'encoder (conta giri) staccati. Probabilmente l'encoder si è usurato durante i test, compromettendo la trasmissione del movimento dall'albero motore all'elemento sensibile dell'encoder.

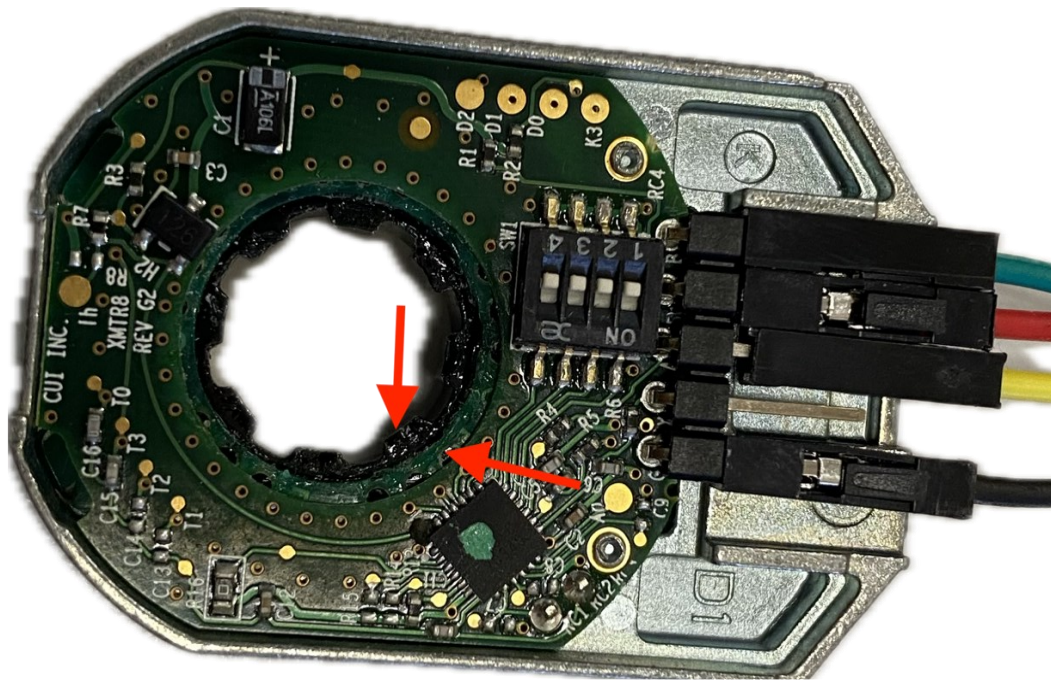


Figura 33: Vista encoder internamente

La soluzione al problema è stata individuata attraverso la riconnessione e il corretto posizionamento dei due componenti interni dell'encoder. Una volta completata la riparazione, sono stati eseguiti ulteriori test per verificare la corretta funzionalità dell'encoder.

Con i componenti dell'encoder adeguatamente riposizionati e fissati, è stato possibile confermare il corretto funzionamento dell'encoder. Un test cruciale è stato condotto per assicurarsi che, durante un giro completo dell'albero motore, il timer in modalità encoder contasse un numero di impulsi pari alla risoluzione dell'encoder moltiplicata per 4 come ci si aspettava.

8.6 Studio del legame duty cycle- velocità (RPM) tramite l'encoder

Questa fase di test è stata fondamentale per valutare la dinamica di funzionamento del sistema di controllo del motore implementato. Durante questa fase, sono stati eseguiti una serie di test mirati per esplorare la relazione tra il duty cycle del segnale PWM inviato all'ESC e la conseguente risposta del motore in termini di velocità RPM (giri del motore al minuto).

L'obiettivo principale è stato comprendere come la regolazione dinamica del Duty-cycle influenzi la velocità del motore, misurata in rotazioni per minuto (RPM). Questa esplorazione ha fornito una visione dettagliata delle relazioni chiave tra il segnale di controllo e la risposta del motore in velocità. Un elemento chiave del mio studio è stato misurare e comprendere come la variazione del Duty-cycle si traduca in cambiamenti nella velocità del motore. Il feedback della velocità fornito dall'encoder è stato quindi cruciale per valutare l'efficacia delle regolazioni apportate e per comprendere come il motore risponda a modifiche dinamiche nel segnale di controllo.

I risultati ottenuti sono stati acquisiti e analizzati utilizzando uno strumento di monitoraggio seriale (Serial Tools) e (Matlab), consentendo una valutazione approfondita delle prestazioni del sistema. Il codice Matlab utilizzato per acquisire e visualizzare i dati dei test è riportato nel paragrafo (7.8)

9 Test e Grafici

Questa sezione presenta un'analisi complessiva dei test eseguiti per esplorare la dinamica tra il duty cycle inviato all'ESC e la variazione della velocità del motore.

Sono stati condotti diversi test, ciascuno caratterizzato da un range di calibrazione specifico e incrementi dinamico di duty cycle Δd (0,05% nei test 1,2 e 3 e 0,20% per il test 4) ogni 5 iterazioni. L'obiettivo principale è comprendere come la calibrazione dell'ESC incida sul controllo della velocità.

L'analisi dettagliata dei risultati ottenuti dai quattro test ha portato a diverse conclusioni fondamentali per la comprensione della dinamica tra il duty cycle inviato all'ESC e la risposta del motore.

1. Il primo grafico cattura l'acquisizione di tutti i punti durante l'incremento del duty cycle di Δd ogni 5 iterazioni. I punti neri mostrano il comportamento completo del sistema, evidenziando variazioni e andamenti specifici. I punti rossi, invece, evidenziano la media di questi punti, fornendo una rappresentazione più chiara e liscia della tendenza complessiva. Questo approccio è particolarmente utile, fornendo una visione più stabile delle prestazioni del motore.
2. Il secondo grafico mette in risalto la media della curva di risposta, evidenziando la transizione graduale delle velocità in risposta agli incrementi del duty cycle. L'approccio di considerare la media degli ultimi tre valori ottenuti con un certo duty cycle, tralasciando i primi 2, risulta essenziale per compensare il ritardo iniziale nella variazione della velocità, offrendo una rappresentazione più accurata delle prestazioni del motore.
3. Le due rette rosse mostrano la tensione della batteria all'inizio e alla fine del test, consentendo di correlare eventuali variazioni nella velocità con la tensione della batteria. Questo aiuta a comprendere l'impatto della tensione sulla dinamica di risposta del motore.

I valori acquisiti in tabelle forniscono un riferimento numerico chiaro e dettagliato, consentendo un'analisi quantitativa approfondita delle variazioni di velocità in relazione agli incrementi del duty cycle. Tuttavia, queste non sono state riportate per la notevole quantità di spazio richiesto, riportando solo i grafici che contengono il medesimo contenuto informativo in versione compatta.

Di seguito, esploriamo le rappresentazioni grafiche con le caratteristiche specifiche per ogni test effettuato.

- **Test1: 5%-15%**

Il primo test è stato condotto per esplorare la relazione tra il duty cycle inviato all'ESC e la variazione della velocità del motore, concentrandosi su un range di calibrazione specifico (5-15%). I dettagli e i risultati del test sono presentati di seguito:

- Tensione Iniziale della Batteria: 16.4V
- Tensione Finale della Batteria: 16.1V
- Range di Calibrazione dell'ESC: 5% - 15%
- Incremento del Duty Cycle: 0.05%

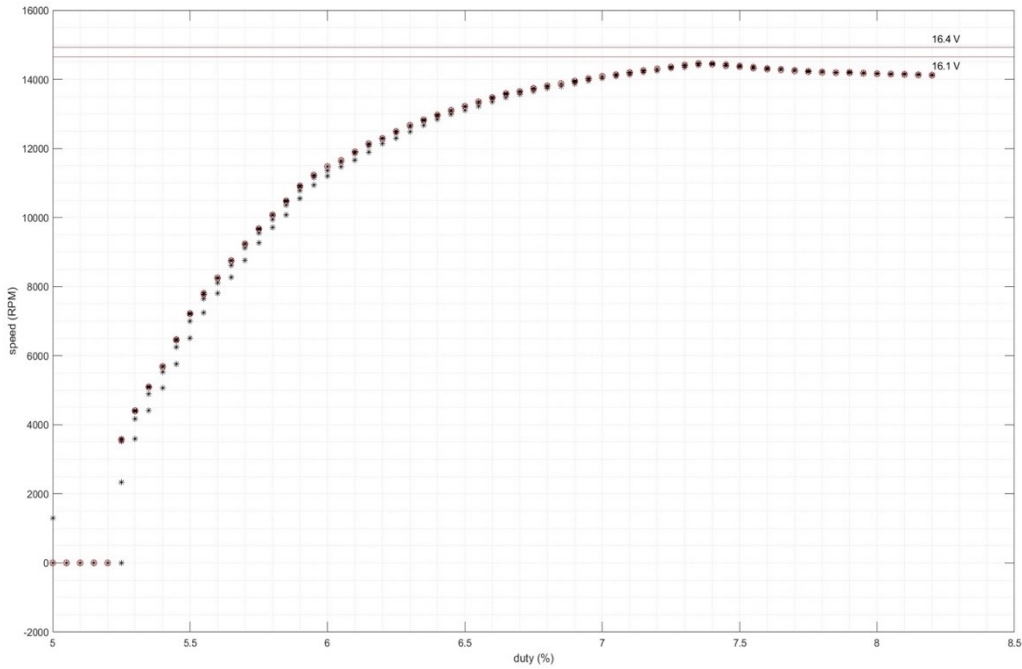


Grafico 1: con tutti i punti acquisiti 5%-15%

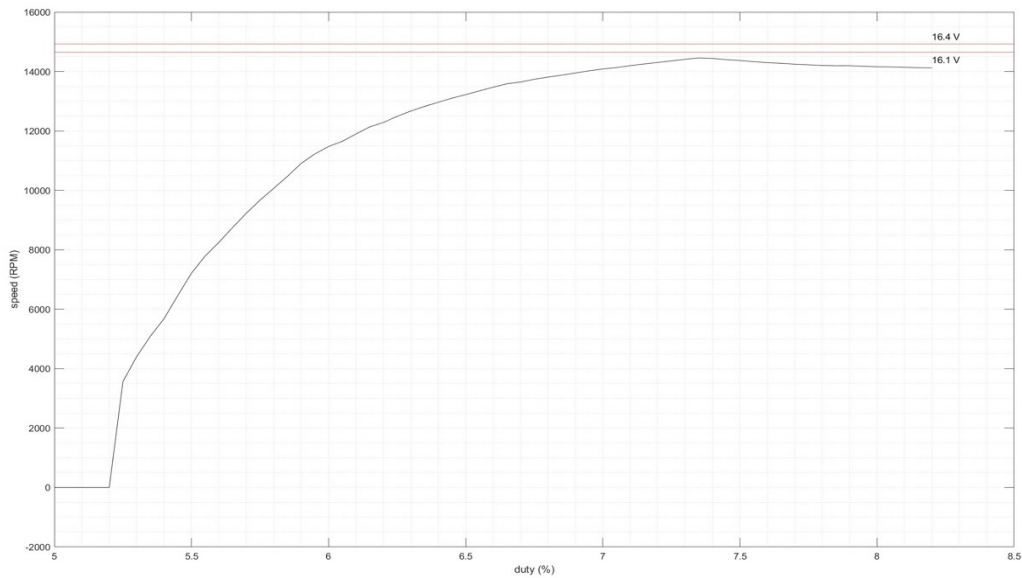


Grafico2 1: Curva del grafico campionata 5%-15%

- **Test2: 5%-30%**

Il secondo test ha coinvolto un range di calibrazione più ampio, focalizzandosi sulle variazioni della velocità del motore in risposta a incrementi del duty cycle. Di seguito sono riportati i dettagli specifici e i risultati chiave emersi durante il test:

- Tensione Iniziale della Batteria: 16.1V
- Tensione Finale della Batteria: 15.9V
- Range di Calibrazione dell'ESC: 5% - 30%
- Incremento del Duty Cycle: 0.05%

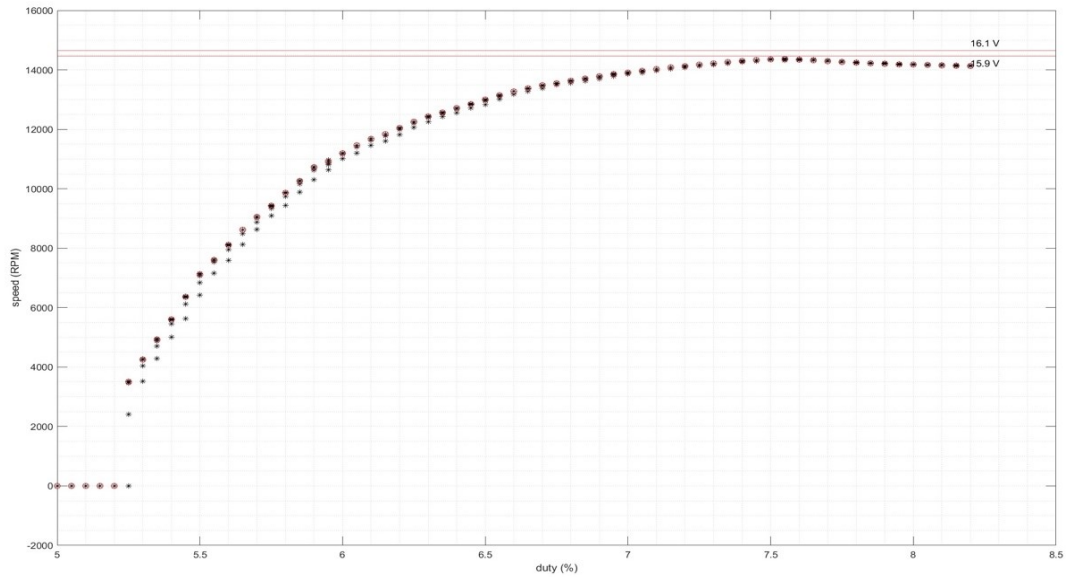


Grafico1 2: Con tutti i punti acquisiti 5%-30%

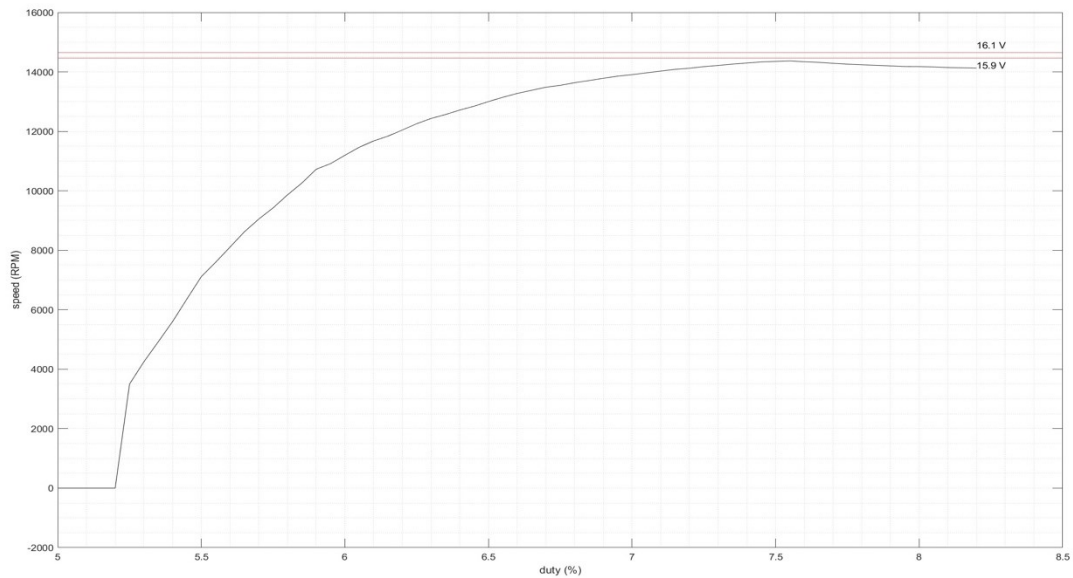


Grafico2 2: Curva del grafico campionata 5%-30%

- **Test3: 4%-14%**

Il terzo test ha coinvolto un range di calibrazione traslato dell'1%, focalizzandosi sulla comprensione delle dinamiche di controllo del motore in risposta a incrementi del duty cycle. Di seguito sono riportati i dettagli specifici e i risultati principali emersi durante il test:

- Tensione Iniziale della Batteria: 16.4V
- Tensione Finale della Batteria: 15.8V
- Range di Calibrazione dell'ESC: 5% - 15%
- Incremento del Duty Cycle: 0.05%

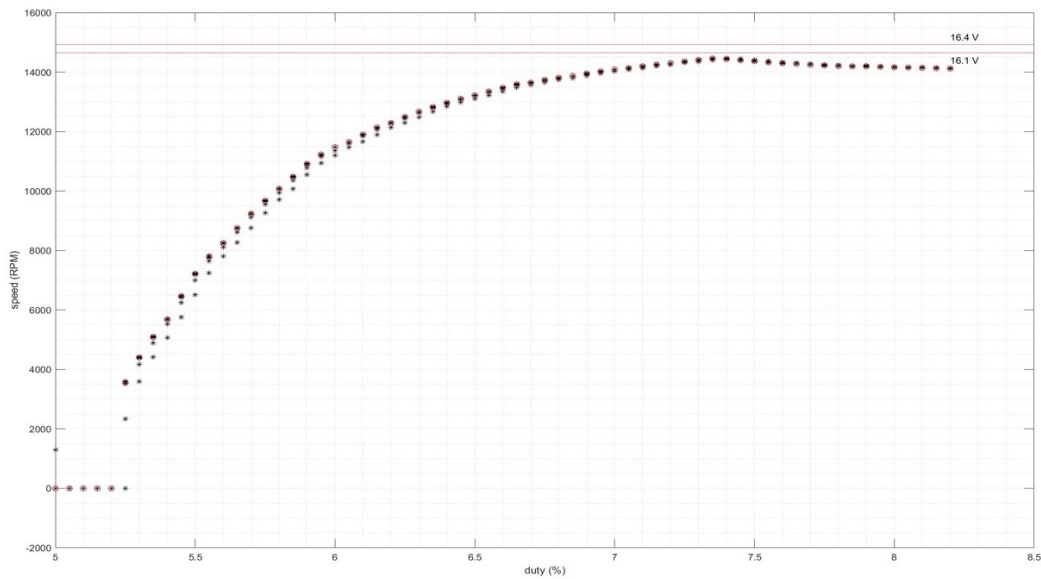


Grafico1 3: Con tutti i punti acquisiti 4%-14%

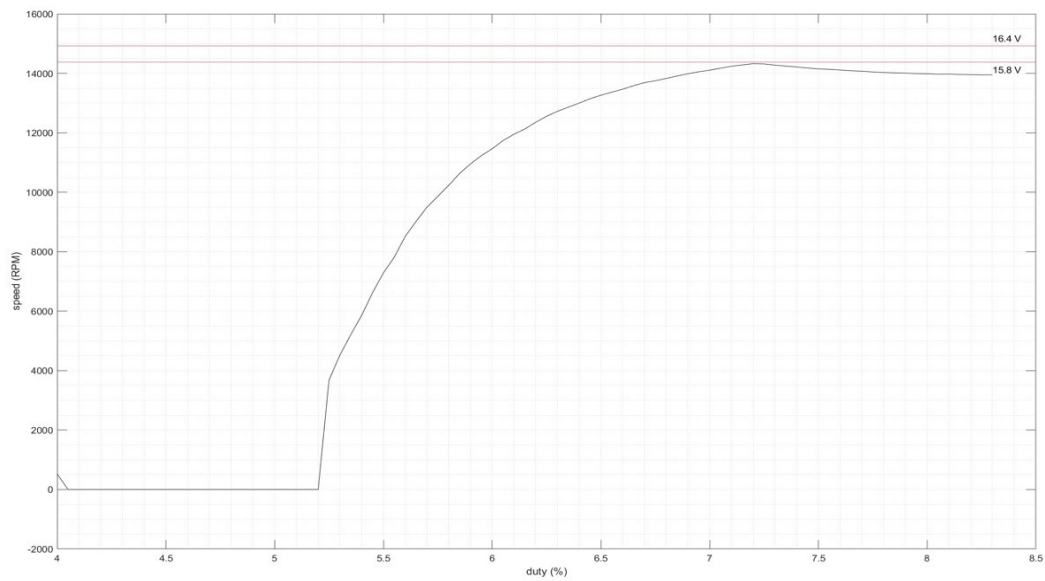


Grafico2 3: Curva del grafico campionata 4%-14%

La procedura di calibrazione non ha prodotto l'effetto atteso, poiché il motore si avvia sempre a 5.2%, indipendentemente dal duty minimo impostato durante la calibrazione. a velocità massima viene raggiunta con un duty all'incirca compreso tra 7.2% e 7.5%, indipendentemente dalla calibrazione.

È emerso un impatto limitato della tensione residua della batteria sulla velocità massima raggiunta. La variazione di tensione non ha comportato variazioni significative nelle prestazioni del motore.

La relazione tra duty cycle e velocità non è lineare ma parabolica. Ciò implica che, a parità di incremento di duty cycle, l'incremento corrispondente nella velocità diventa via via minore.

Il motore impiega del tempo per raggiungere la nuova velocità, nonostante i piccoli incrementi di duty cycle. I primi valori di velocità sono più bassi, e quindi, si è scelto di considerare la media degli ultimi tre valori come valore di velocità per ottenere una stima più accurata come viene mostrato nei secondi grafici denominati (Curva del grafico campionata).

9.1 Studio dei limiti del brushless

In questo ultimo test, l'obiettivo principale è stato replicare e approfondire quanto osservato nella sezione (8.2), focalizzandosi sulla determinazione del duty cycle oltre il quale il motore si spegne. In questo test si è lavorato per un duplice scopo, identificare il valore specifico di duty cycle a cui si verifica lo spegnimento del motore e analizzare la variazione di velocità in prossimità di questo valore critico.

- **Test4: 5%-15%**
 - Tensione Iniziale della Batteria: 16.2V
 - Tensione Finale della Batteria: 15.9V
 - Range di Calibrazione dell'ESC: 5% - 15%
 - Incremento del Duty Cycle: 0.2%

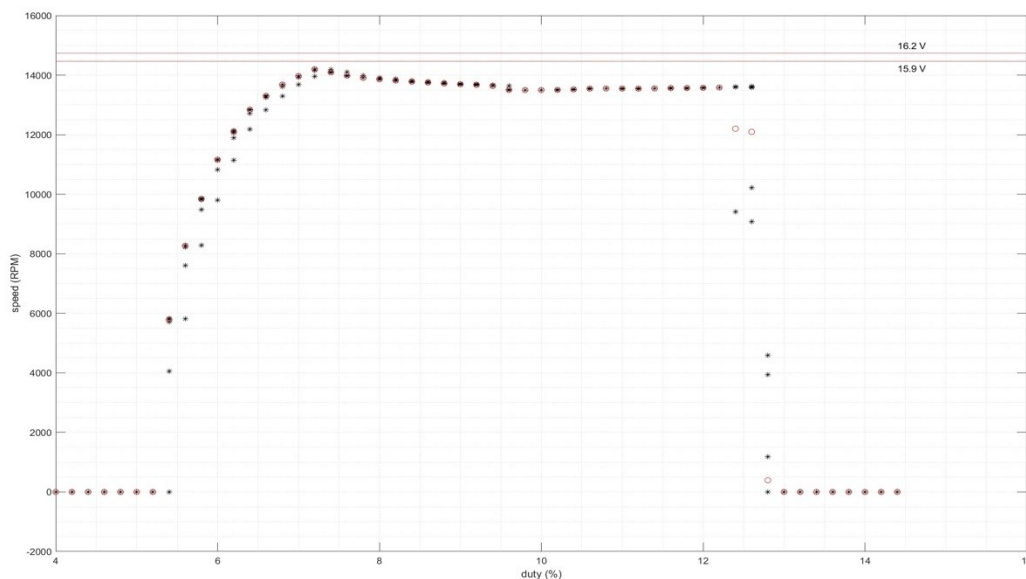


Grafico 4: Con tutti i punti acquisiti 5%-15%

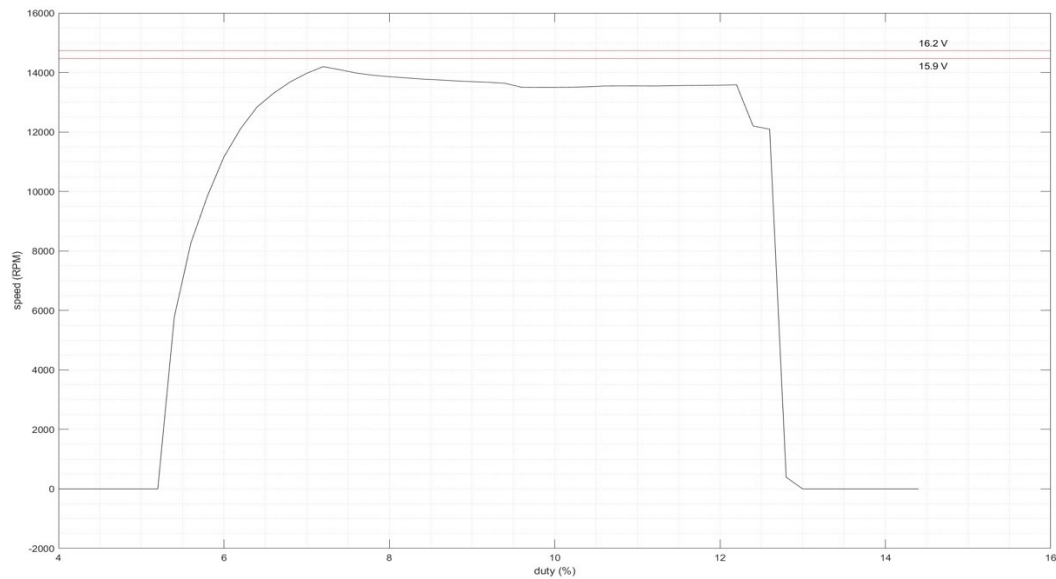


Grafico2 4: Curva del grafico campionata 5%-15%

Questo test è stato progettato con una strategia di incremento più veloce del duty cycle (0.2%), consentendo di identificare più velocemente il punto in cui il motore raggiunge il limite e si spegne. La variazione della tensione della batteria durante il test fornisce ulteriori informazioni sulla stabilità del sistema in condizioni di tensione variabile.

I risultati di questo test contribuiranno a consolidare la comprensione dei limiti di funzionamento del motore in termini di duty cycle e forniranno ulteriori dettagli sulla dinamica di variazione della velocità in prossimità di questi punti critici.

CONCLUSIONI E SVILUPPI FUTURI

Questo progetto ha (parzialmente) raggiunto l'obiettivo che si era prefissato, ovvero la comprensione della catena batteria-ESC-brushless-elica adibita alla propulsione di un drone quadrirotore. Tale comprensione è infatti fondamentale per poter poi implementare un controllore che sia in grado di stabilizzare il drone. In particolare, la relazione più rilevante che è stata approfondita è quella che sussiste tra il comando PWM dell'ESC e la velocità del brushless da essa controllato dal momento che l'algoritmo di controllo di un quadricottero richiede di controllare e modificare le velocità dei singoli motori brushless (Paragrafo 1.2).

Per raggiungere questo obiettivo si è partiti dallo studio delle relazioni matematiche che sussistono tra questi componenti in modo da comprendere il funzionamento del sistema. Combinando queste formule con i dati tecnici forniti dai produttori dei componenti utilizzati in fase di test, è stato possibile prevedere il comportamento del sistema e quindi verificare che tutto avvenisse secondo le aspettative. In sintesi, lo studio condotto ha permesso di concludere che: la batteria fornisce la tensione e la corrente necessarie, l'ESC controlla la velocità del motore brushless attraverso il segnale PWM, il motore brushless risponde alla tensione applicata e le eliche generano spinta in base al loro passo. Tutti questi componenti lavorano insieme per generare la spinta per il veicolo.

1. Calibrazione duty min e max

In contrasto con le aspettative iniziali, l'esperimento ha rivelato la presenza di limiti specifici nel range di duty cycle che può essere impostato durante la fase di calibrazione. In particolare, sono emerse soglie, sia minime sia massime, al di fuori delle quali la calibrazione non riesce con successo. Questo fenomeno ha evidenziato la complessità e la sensibilità del processo di calibrazione del duty cycle.

L'osservazione di soglie sia minime sia massime indica che la calibrazione richiede un intervallo di duty cycle specifico per essere completata con successo.

La presenza di limiti imprevisti al di fuori delle aspettative iniziali suggerisce una variabilità nel comportamento del sistema che potrebbe essere influenzata da fattori specifici del contesto operativo.

Questi risultati trovano riscontro nel (Paragrafo 8.2), dove è stato precedentemente osservato un comportamento analogo durante il processo di calibrazione.

Queste nuove informazioni sollevano importanti interrogativi riguardo alle dinamiche interne del sistema e sottolineano l'importanza di ulteriori indagini per comprendere appieno i limiti di calibrazione del duty cycle. Questa scoperta inaspettata aggiunge un livello di complessità al processo di configurazione del sistema, richiedendo un'analisi più approfondita per identificare le cause di tali limiti e determinare eventuali misure correttive o ottimizzazioni.

2. Limiti di Duty Cycle

I test svolti hanno mostrato che, a prescindere dal range impostato in fase di calibrazione il motore impone dei limiti sia al duty cycle minimo in corrispondenza del quale il motore inizia a ruotare, sia come duty massimo oltre il quale la velocità smette prima di aumentare, fino ad arrivare ad una soglia oltre cui il motore si spegne, come spiegati nel paragrafo (8.2) per il setup1 e nel capitolo (9).

3. Duty Cycle Minimo:

Un duty cycle inferiore al 5% è stato identificato come insufficiente per avviare il motore. Questo risultato sottolinea l'importanza cruciale di definire un range di duty cycle ottimale durante la calibrazione dell'ESC. La limitazione al di sotto del 5% può essere interpretata come una misura di sicurezza, finalizzata a prevenire avvii accidentali o danneggiamenti al motore.

4. Duty Cycle Massimo:

Oltre il 15% di duty cycle massimo, si è osservato un comportamento in cui la velocità del motore si annulla indipendentemente dal valore del duty cycle. Questo limite superiore riflette un compromesso accurato tra sicurezza operativa e prestazioni ottimali, prevenendo potenziali surriscaldamenti o sovraccarichi.

5. Limite di Spegnimento:

Durante i test, è stato notato che l'ESC consente la calibrazione con duty cycle superiore al limite definito. Tuttavia, il motore non raggiunge mai il massimo consentito, il che è attribuibile alle specifiche del motore stesso. Questo aspetto sottolinea che, nonostante l'ESC offra la flessibilità di calibrare con duty cycle più elevati, è il motore che, in base alle sue specifiche, determina il reale raggiungimento del massimo del duty cycle.

Oltre una determinata soglia di duty cycle, il motore si spegne, come dimostrato nel paragrafo (9.1). Questo limite rappresenta un punto critico oltre il quale ulteriori aumenti del duty cycle non contribuiscono alla velocità del motore, ma comportano invece lo spegnimento del motore. Questa caratteristica evidenzia la necessità di definire attentamente il range di duty cycle per evitare situazioni di spegnimento indesiderato.

Tuttavia, al di là di questo punto, il motore ha raggiunto il massimo dei suoi giri al minuto (RPM), influenzato principalmente dal valore del KV del motore brushless. Ciò è avvenuto indipendentemente dal fatto che il duty cycle definito raggiungesse o meno il suo valore massimo. Inoltre, l'applicazione della formula teorica di velocità (RPM) = $KV \times T$, come spiegata nel paragrafo (3.3), ha confermato la validità teorica del nostro approccio, contribuendo a consolidare la coerenza tra le aspettative teoriche e i risultati sperimentali.

Probabilmente questi limiti sono legati a meccanismi di sicurezza che impediscono l'avvio o fermano il motore. La presenza di queste soglie si è rivelata essere un elemento critico per la sicurezza operativa del sistema dal momento che fa perdere il controllo del motore.

In conclusione, emerge chiaramente che la corretta calibrazione dell'ESC costituisce un elemento cruciale per assicurare prestazioni affidabili e sicure dei motori brushless. Trovare un equilibrio tra la potenza erogata e la sicurezza è essenziale, e la definizione di un range di duty cycle ottimale si presenta come una strategia chiave in questo processo. Da questa analisi, si può dedurre che l'ESC è più flessibile riguardo ai duty cycle massimo rispetto a quello minimo. Questo significa che l'ESC offre la possibilità di lavorare con un ampio range di duty cycle, consentendo così la scelta di diversi motori con diverse specifiche. Tuttavia, è importante sottolineare che è il motore stesso a porre dei limiti a tale flessibilità, determinando il reale raggiungimento del massimo del duty cycle che può ricevere in base alle sue specifiche. In definitiva, l'ESC fornisce

una piattaforma versatile, adattabile a diverse configurazioni di motori, mentre il motore, con le sue caratteristiche uniche, stabilisce i limiti operativi effettivi.

6. Legame non lineare tra Duty-RPM

Durante la fase di sperimentazione, i test effettuati hanno chiaramente convalidato l'influenza diretta del duty cycle del segnale PWM sull'ESC e, di conseguenza, sulla potenza erogata alle fasi del motore, modulando la sua velocità di rotazione.

I test hanno chiaramente evidenziato che variazioni nel duty cycle del segnale PWM inviato all'ESC hanno un impatto diretto sulla potenza erogata alle fasi del motore. Un aumento del duty cycle si traduce in un incremento della potenza fornita al motore, influenzando la sua velocità di rotazione.

La relazione tra duty cycle e velocità del motore è stata confermata come non lineare ma parabolica. Ciò significa che, a parità di incremento di duty cycle, l'incremento corrispondente nella velocità diventa progressivamente minore.

7. Sviluppi futuri

Per quanto riguarda infine gli sviluppi futuri, si prevede quanto segue:

- Esplorare ulteriormente gli aspetti teorici relativi alle dinamiche dei motori brushless e dei regolatori elettronici di velocità (ESC). Un focus particolare potrebbe essere dato alle caratteristiche avanzate di controllo e alle modalità operative degli ESC.
- Ripetere i test in diverse condizioni operative per confermare la robustezza dei risultati ottenuti. La configurazione meccanica e la tensione della batteria dovrebbero essere esaminate per valutare la coerenza delle prestazioni del sistema.
- Approfondire l'analisi dei comportamenti anomali riscontrati durante i test, in particolare quelli legati all'impatto della procedura di calibrazione. Studiare le ragioni sottostanti a tali fenomeni e sviluppare soluzioni o miglioramenti alla procedura di calibrazione per garantire una maggiore precisione.
- Progettare e implementare un nuovo setup sperimentale dotato di sensori avanzati per misurare con precisione la spinta dell'elica e monitorare la tensione e la corrente delle fasi del motore. L'obiettivo è acquisire dati più dettagliati per una valutazione completa delle prestazioni del sistema e una comprensione approfondita delle variabili coinvolte.
- Esplorare l'implementazione di algoritmi di controllo più avanzati, come il controllo PID (Proporzionale-Integrativo-Derivativo) o altri algoritmi di controllo adattivo. Ciò potrebbe consentire un controllo più preciso e dinamico del motore in risposta alle variazioni di carico e alle condizioni operative.
- Esplorare l'integrazione di sensori avanzati, come giroscopi o accelerometri, per migliorare la percezione e la risposta del sistema alle variazioni dell'ambiente. Ciò potrebbe contribuire a una maggiore precisione nella regolazione del motore.
- Provare diverse configurazioni di encoder o considerare l'utilizzo di encoder di alta precisione per migliorare la precisione delle letture durante l'implementazione del feedback.

- Approfondire lo studio delle caratteristiche di carico e delle dinamiche del motore in diversi scenari operativi. Questo potrebbe includere l'analisi del comportamento del motore in presenza di carichi variabili e la definizione di strategie di controllo ottimali per ciascuna situazione.
- Replicare i test svolti in questo lavoro con il nuovo setup in modo da confermare quanto appreso in termini di relazione tra calibrazione, armatura e controllo della velocità del motore;

Ulteriori sviluppi futuri potrebbero invece riguardare la realizzazione di un ulteriore setup che sia in grado di misurare contemporaneamente non solo la velocità di rotazione del motore ma anche la tensione della batteria, la tensione di alimentazione delle fasi del motore e la spinta generata dall'elica in modo da poter monitorare tutte le grandezze coinvolte nella catena. Un setup di questo tipo potrebbe essere utilizzato anche per studiare il comportamento del sistema in presenza di carichi dinamici.

Bibliografia

- [1] YUMPU, «TURNIGY Manual for Brushless Motor Speed Controller,» [Online]. Available: <https://www.yumpu.com/en/document/view/23102122/turnigy-manual-for-brushless-motor-speed-controller>.
- [2] CUDEVICES, [Online]. Available: <https://www.cuidevices.com/product/resource/amt10.pdf>.
- [3] F. SPIRANTI, F. SPIRANTI e S. CENINI, «AF DRONES,» [Online]. Available: <https://afdrones.it/che-cose-lesc/>.
- [4] CONTROLLERSTECH SHOP & LEARN, «How to interface BLDC motor with STM32,» [Online]. Available: <https://controllerstech.com/how-to-interface-bldc-motor-with-stm32/>.
- [5] Coderdojotrento.it/arduino3, «Arduino – ESC e motori droni Speciale «Festivolare 2017»,» [Online]. Available: <https://static.coderdojotrento.it/arduino/tutorial/3-brushless/tutorial-3-arduino-brushless.pdf>.
- [6] HowToMechatronics , «Arduino Brushless Motor Control Tutorial | ESC | BLDC,» [Online]. Available: https://howtomechatronics.com/tutorials/arduino/arduino-brushless-motor-control-tutorial-esc-bldc/#google_vignette.
- [7] DeepBlueMbedded, «STM32 EcoSystem (Development Environment) Setup,» [Online]. Available: <https://deepbluembedded.com/stm32-ecosystem-development-environment-setup/>.
- [8] TYTO ROBOTICS, «What is an Electronic Speed Controller & How Does an ESC Work,» [Online]. Available: <https://www.tytorobotics.com/blogs/articles/what-is-an-esc-how-does-an-esc-work#:~:text=The%20role%20of%20the%20ESC,translated%20into%20changes%20in%20speed..>
- [9] Hobbyking, «Turnigy-plush-32-brushless-speed-controller-programming,» [Online]. Available: https://hobbyking.com/en_us/turnigy-plush-32-brushless-speed-controller-programming-card-rev1-1-0.html.