

UNIVERSITÀ POLITECNICA DELLE MARCHE
INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE
Relazione catena
BATTERIA-ESC-BRUSHLESS



Corso di
LABORATORIO DI AUTOMAZIONE
Anno accademico 2023-2024

Studenti:
Luca D'Ambrosio
Luca Tassoni
Niccolò Tassoni

Professore:
Andrea Bonci
Dottorando:
Alessandro Di Biase



Dipartimento di Ingegneria dell'Informazione

Indice

Introduzione	2
1 Sistema	4
1.1 Quadrirotore	4
1.2 Catena Batteria-ESC-Brushless	4
2 Hardware	6
2.1 Scheda STM32H745 Nucleo-144	6
2.2 Batteria LiPo Turnigy A-Spec 2600 mAh 4S	6
2.3 Banco di Alimentazione	7
2.4 ESC Turnigy Plush 30A	9
2.4.1 Programmazione ESC	9
2.4.2 Procedura di calibrazione	10
2.5 Motore Brushless Turnigy D3536/9 910KV	11
2.6 AMT102-D2048-I8000-S	12
2.7 AMPROBE 5XP-A	15
2.8 Schema dei collegamenti	15
2.8.1 Setup batteria	15
2.8.2 Setup banco	17
3 Software	18
3.1 STM32CubeIDE	18
3.2 Putty	19
3.3 Serial Tools	19
3.4 MatLab	20
3.5 Diagramma di flusso	21
3.6 Gestione singoli componenti	23
3.6.1 Gestione calibrazione ESC	23
3.6.2 Gestione Trimmer	26
3.6.3 Gestione Encoder	28
3.6.4 Gestione duty incrementale	30
3.7 Funzionamento complessivo	32
3.8 Codice Matlab	32
4 Test e risultati	34
4.1 Test con trimmer duty	34
4.2 Test con banco di alimentazione duty/velocità	34
4.2.1 5-10%	34
4.2.2 8-12%	35
4.2.3 5-15%	36
4.3 Test con banco di alimentazione velocità/tensione	37

4.3.1	Tensione=16.8V	37
4.3.2	Tensione=15.0V	38
4.3.3	Tensione=13.0V	39
4.4	Spegnimento del motore	40
4.4.1	Cut-Off	42
4.4.2	ESC Protection Mode	42
4.4.3	Tensioni e Correnti elevate per il motore	42
4.4.4	Conclusione	43
	Conclusioni e sviluppi futuri	43

Introduzione

Questa relazione ha come obiettivo l'approfondimento del controllo di un sistema tramite ESC. Si concentra sulla connessione tra 3 componenti: batteria, ESC (Electronic Speed Controller) e motore.

L'interesse di questa relazione è rivolto a comprendere i limiti e il campo di applicazione dei singoli componenti, per poter sviluppare un controllo di un motore più preciso. In particolare ci siamo concentrati sui comportamenti anomali osservati nelle precedenti relazioni.

Il nostro lavoro è diviso in una prima fase di studio in cui abbiamo approfondito gli aspetti teorici del nostro sistema, in una seconda fase di test per verificare i comportamenti anomali ottenuti, e in una terza fase di ricerca delle possibili cause di tali comportamenti.

Nel nostro caso specifico abbiamo lavorato sul sistema di propulsione di un quadrirotore, collaborando con i nostri colleghi che si sono occupati del controllo della stabilità in volo, l'obiettivo è quello di sviluppare un sistema stabile e controllabile.

Nei capitoli seguenti vengono presentati il sistema studiato, l'hardware, il software, e i test da noi effettuati.

1 Sistema

In questo capitolo spieghiamo brevemente il sistema complessivo da noi analizzato e la relazione tra i suoi componenti.

1.1 Quadrirotore



Un quadrirotore è una tipologia di veicolo aereo multirrotore, presenta quattro motori disposti a croce che permettono il decollo verticale. 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'hobbyismo 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. [1]

1.2 Catena Batteria-ESC-Brushless

Nel nostro caso di studio ci siamo concentrati sulla catena batteria-ESC-motore brushless e sui limiti applicativi dei singoli componenti. In un normale drone quadrirotore la batteria è collegata a quattro ESC a loro volta collegati ad altrettanti motori. Per semplicità abbiamo considerato uno solo di questi collegamenti, per poter ottenere informazioni che possano contribuire a migliorare il comportamento del sistema complessivo.

Anche se negli utilizzi pratici l'ESC viene comandato tramite radiocomando noi abbiamo utilizzato la scheda STM in modo da poter analizzare più nel profondo il comportamento del sistema.

Si ha che la batteria è la fonte della tensione in ingresso all'ESC, essendo una batteria offre tensione continua. Il compito dell'ESC è quello di gestire la tensione in ingresso dalla batteria, generando delle tensioni alternate per l'alimentazione trifase del motore. Il collegamento tra l'ESC e la scheda STM è fondamentale perché tramite il microcontrollore ci è possibile gestire il segnale PWM, comunemente usato per il controllo dei motori elettrici. L'ESC, infatti, è dotato del BEC (Battery Elimination Circuit) che permette di collegarlo direttamente con la scheda.

Il motore Brushless trifase è diviso in due componenti principali: lo statore e il rotore.

Nello statore alloggia il circuito di alimentazione, mentre nel rotore sono posizionati dei magneti permanenti. In generale si utilizza un' alimentazione trifase perchè consente di avere prestazioni più lineari e una coppia più regolare. All' ESC è affidato il compito di gestire la commutazione delle correnti, che variano l'orientamento del campo magnetico nelle espansioni polari dello statore, permettendo la rotazione.

2 Hardware

2.1 Scheda STM32H745 Nucleo-144

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. Inoltre assicura l'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). $145 \mu\text{A}/\text{MHz} @ \text{VDD}$ tipici = 3,3 V e 25 °C in modalità Run (periferiche spente) e SMPS 2,43 μA 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

2.2 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



Figure 2.1: 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.

2.3 Banco di Alimentazione

Nel progetto è stato utilizzato il banco di alimentazione, fornito dal laboratorio, con le seguenti caratteristiche [2]:



Figure 2.2: Banco di Alimentazione

SPECIFICATIONS																																						
OUTPUT MODE	Voltage	GPS-4303				GPS-4251				GPS-3303		GPS-2303																										
		CH1	CH2	CH3	CH4	CH1	CH2	CH3	CH4	CH1	CH2	CH3	CH1	CH2																								
		0 ~ 30V	2.2 ~ 5.2V	8 ~ 15V	0 ~ 25V	3 ~ 6V	8 ~ 15V	0 ~ 30V	5V Fixed	0 ~ 30V																												
		0 ~ 3A	1A Max.	1A Max.	0 ~ 0.5A	2.5A Max.	1A Max.	0 ~ 3A	3A Max.	0 ~ 3A																												
		0 ~ 60V	-----	0 ~ 50V	-----	0 ~ 60V	-----	0 ~ 60V	-----	0 ~ 60V																												
		0 ~ 6A	-----	0 ~ 1A	-----	0 ~ 6A	-----	0 ~ 6A	-----	0 ~ 6A																												
CONSTANT VOLTAGE OPERATION (CH1, CH2)		Line Regulation	$\leq 0.01\% + 3mV$																																			
		Load Regulation	$\leq 0.01\% + 3mV$ (rating current < 3A)																																			
		Ripple & Noise	$\leq 0.02\% + 5mV$ (rating current > 3A)																																			
		Recovery Time	$\leq 1mVrms, 5Hz \sim 1MHz$																																			
			$\leq 100\mu S$ (50% Load change, Minimum load 0.5A)																																			
CONSTANT CURRENT OPERATION (CH1, CH2)		Line Regulation	$\leq 0.2\% + 3mA$																																			
		Load Regulation	$\leq 0.2\% + 3mA$																																			
		Ripple & Noise	$\leq 3mA rms$																																			
TRACKING OPERATION (CH1, CH2)		Tracking Error	$\leq 0.5\% + 10mV$ of the CH1																																			
		Series Regulation	$\leq 300mV$																																			
		Load Regulation	$\leq 0.01\% + 3mV$																																			
		Ripple & Noise	$\leq 2mVrms, 5Hz \sim 1MHz$																																			
GPS-4303 2.2V ~ 5.2V OUTPUT (CH3) GPS-3303 5V FIX OUTPUT (CH3) GPS-4251 3V~6V OUTPUT(CH3)			Line Regulation	$\leq 5mV$																																		
			Load Regulation	$\leq 15mV$																																		
			Ripple & Noise	$\leq 2mVrms, 5Hz \sim 1MHz$																																		
			Current Output	GPS-4303: 1A, GPS-3303: 3A, GPS-4251: 2.5A																																		
GPS-4303 8V ~ 15V OUTPUT (CH4) GPS-4251 8V-15V OUTPUT (CH4)			Line Regulation	$\leq 5mV$																																		
			Load Regulation	$\leq 10mV$																																		
			Ripple & Noise	$\leq 2mVrms, 5Hz \sim 1MHz$																																		
			Current Output	1A																																		
METER		Digital	3 digits 0.5" LED display GPS-4303/4251/3303 Out ON Accuracy $\pm (0.5\% \text{ rdg} + 2 \text{ digits})$ GPS-4303/3303 Out OFF Accuracy $\pm (0.5\% \text{ rdg} + 8 \text{ digits})$ GPS-2303 Accuracy $\pm (0.5\% \text{ rdg} + 2 \text{ digits})$																																			
INSULATION		Chassis and Terminal Chassis and AC Cord	$\geq DC 500V / 20M\Omega$ $\geq DC 500V / 30M\Omega$																																			
POWER SOURCE		AC 100V/120V/220V($\pm 10\%$)/230V($\pm 10\% \sim -6\%$), 50/60Hz																																				
ACCESSORIES		Power cord x 1, Instruction manual x 1 GPS-4303 : Test lead GTL-104 x 2, GTL-105 x 2 or GTL-203 x 2, GTL-204 x 2 GPS-4251 : Test lead GTL-105 x 4 or GTL-203 x 4 GPS-3303 : Test lead GTL-104 x 2, GTL-105 x 1 or GTL-203 x 1, GTL-204 x 2 GPS-2303 : Test lead GTL-104 x 2 or GTL-204 x 2																																				
DIMENSIONS & WEIGHT		255(W) x 145(H) x 265(D) mm, Approx. 7 kg (GPS-4303/3303/2303) ; Approx. 6.3kg (GPS-4251)																																				

Figure 2.3: Alimentatore da banco [2]

2.4 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
BEC	5V 2A ver.3,1
Fonte Manuale	Turnigy Plush 30A



Figure 2.4: Turnigy Plush 30A

2.4.1 Programmazione 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) arresta il motore istantaneamente, altrimenti rallenta naturalmente e dopodiché si arresta a causa della sua inerzia.
- **Tipologia di Cut-Off (Soft-cut/Cut-off)**: se settato su “Cut-off”, consente di ridurre immediatamente la potenza fornita al motore una volta raggiunta la tensione di cut-off (vedi punto successivo), altrimenti la riduzione di potenza è graduale.
- **Tensione di Cut-Off (Low/Middle/High)**: definisce la tensione minima di alimentazione dell’ESC (che corrisponde alla tensione erogata dalla batteria), al di sotto della quale il motore non viene alimentato. Per le 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.

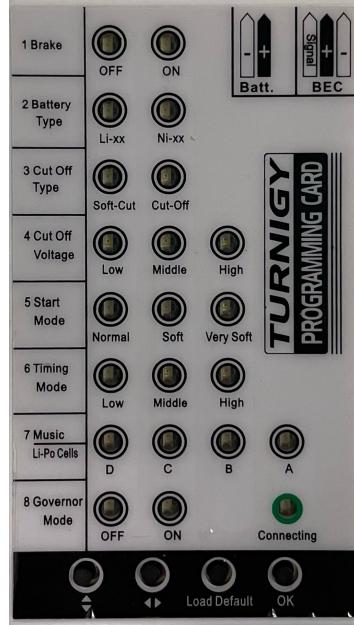


Figure 2.5: Scheda di programmazione di un ESC

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.

2.4.2 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.

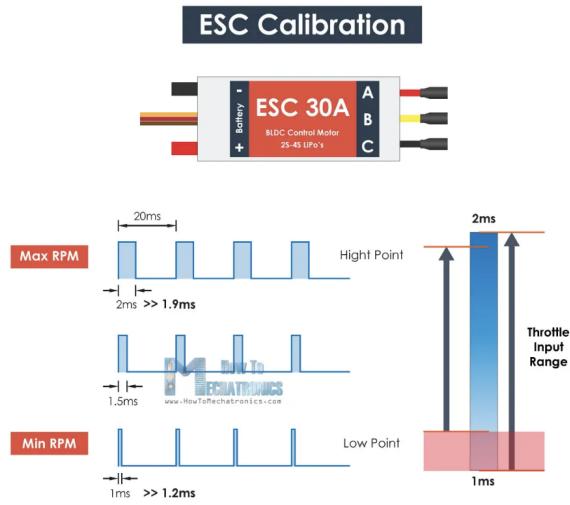


Figure 2.6: figura 10: Effetto procedura di calibrazione

2.5 Motore Brushless Turnigy D3536/9 910KV

In questo progetto è stato utilizzato il motore turnigy D3536/9 910KV.

Nella modellistica è comune l'utilizzo di motori brushless, infatti rispetto ad altre tipologie di motori questi hanno vari vantaggi, ad esempio un minor peso a parità di potenza. Nel nostro caso il motore è un brushless trifase outrunner, ovvero dotato di uno statore fisso posto internamente al rotore. Questa configurazione permette una coppia sostenibile elevata, ed è molto utilizzata nelle applicazioni di aeromodellismo. Questo motore è anche sensorless, ovvero non necessita di sensori per determinare la posizione angolare del rotore, rendendo più facile il controllo dell'alimentazione delle tre fasi.

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.

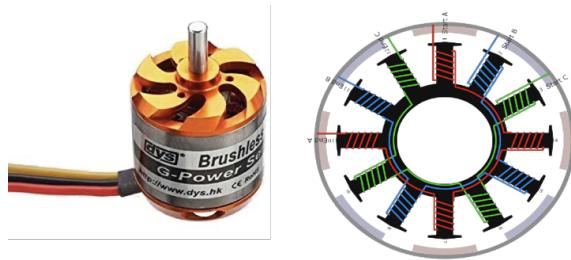


Figure 2.7: Motore brushless Turnigy D3536/9 910KV

Le caratteristiche tecniche del motore utilizzato sono le seguenti:

KV	910KV
Fasi	3
Massima corrente	25.5 A
Potenza massima	370W a 15V
Peso	102 kg
Fasi	3
Batteria	2 ~ 4 Cell /7.4 ~ 14.8V
Diametro del pozzo	5mm
No corrente di carico	1.5 A
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 910 RPM/V significa che se lo alimentiamo con una cella LiPo da 3.7V il motore effettuerà 3367 rotazioni al minuto (910×3.7).

2.6 AMT102-D2048-I8000-S



Figure 2.8: encoder ATM 102

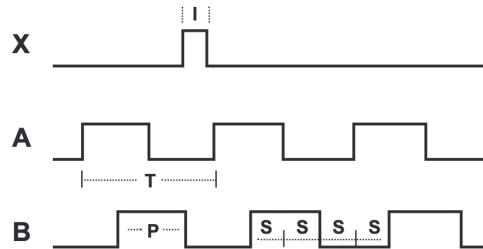
In figura è mostrato l'encoder utilizzato per la parte dei test. Gli encoder sono sensori di posizione in grado di rilevare variazioni angolari. Nel nostro progetto abbiamo utilizzato un encoder per poter determinare le prestazioni del nostro motore. Nel nostro caso specifico l'encoder ATM10 è modulare, incrementale, capacitivo ed in quadratura.

1. È modulare perchè tramite uno switch è possibile impostare il suo PPR.

2. È incrementale perchè fornisce una posizione relativamente a quella precedentemente misurata.
3. È in quadratura perchè il segnale è prodotto da due fasi sfasate tra di loro di 90° gradi. Questo comporta una maggiore risoluzione e la possibilità di capire se la rotazione avviene in senso orario o antiorario. 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.

Oltre al valore di PPR impostato la risoluzione dell'encoder dipende anche dal parametro K, infatti dalla teoria abbiamo che $\Delta\Theta \equiv \frac{360}{K \cdot PPR}$, dove K assume i valori 1,2,4. 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. |



The following parameters are defined by the resolution selected for each encoder. The encoders resolution is listed as Pulses Per Revolution (PPR), which is the number of periods (or high pulses) over the encoders revolution.

Parameter	Description	Expression	Units	Notes
PPR	resolution		Pulses Per Revolution	This is the user selected value and the format all resolutions are listed in
CPR	counts	PPR x 4	Counts Per Revolution	This is the number of quadrature counts the encoder has
T	period	360/R	mechanical degrees	
P	pulse width	T/2	mechanical degrees	
S	A/B state width	T/4	mechanical degrees	This is the width of a quadrature state
I	index width	T/4	mechanical degrees	The width of a once per turn index is the state width for A & B lines

Figure 2.9: Segnali di quadratura, parametri encoder[3]

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

1 = On, 0 = Off

DIP switch:
Example setting: 500 PPR

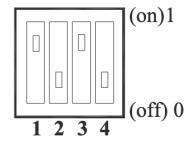


Figure 2.10: Impostazioni della risoluzione[3]

2.7 AMPROBE 5XP-A

In questo progetto abbiamo utilizzato un multmetro AMPROBE 5XP-A.



Figure 2.11: Multimetro

I multimetri sono strumenti di misura che permettono di misurare tensioni e correnti continue e discontinue. Di seguito riportiamo le sue specifiche tecniche: [4]

Voltage		
DC Volts		
Ranges	200 mV, 2 V, 20 V, 200 V, 1000 V	$\pm (1.0 \% \text{ rdg} + 1 \text{ dgt})$
AC Volts (45 Hz to 500 Hz)		
Ranges	200 mV, 2 V, 20 V, 200 V, 750 V	$\pm (1.5 \% \text{ rdg} + 5 \text{ dgt})$
Resolution	100 μV	
Input Impedance	10 M Ω	
Overload Protection	200 mV: 1000 V DC or 750 V AC RMS (3 minutes) All other ranges: 1000 V DC or 750 V AC RMS	
Current		
DC Current		
Ranges	200 μA , 2 mA, 20 mA, 200 mA	$\pm (1.5 \% \text{ rdg} + 1 \text{ dgt})$
AC Current (45 Hz to 500 Hz)		
Ranges	200 μA , 2 mA, 20 mA, 200 mA	$\pm (2.0 \% \text{ rdg} + 5 \text{ dgt})$
Resolution	0.1 μA	
Input Protection	315 mA/1000 V fast blow ceramic fuse 6.3 mm x 32 mm on mA input	
Burden Voltage	300 mV max. (1 V max. on 200 mA range)	
Resistance	200 Ω , 2 k Ω , 20 k Ω , 200 k Ω	$\pm (1.0 \% \text{ rdg} + 4 \text{ dgt})$
	2 M Ω	$\pm (1.5 \% \text{ rdg} + 4 \text{ dgt})$
	20 M Ω	$\pm (2.0 \% \text{ rdg} + 5 \text{ dgt})$
Resolution	100 m Ω	
Open Circuit Voltage	0.3 V DC typical, (3.0 V DC on 200 Ω range)	
Overload Protection	1000 V DC or 750 V AC RMS	
Continuity		
Audible Indication	75 Ω	$\pm 25 \Omega$
Response Time	100 ms	
Overload Protection	1000 V DC or 750 V AC RMS	

Figure 2.12: Data Sheet Multimetro[4]

2.8 Schema dei collegamenti

2.8.1 Setup batteria

Nello schema dei collegamenti in Figura 2.13 vengono riportati i principali collegamenti del sistema, la catena batteria, ESC, e motore brushless, la scheda STM e l'encoder.

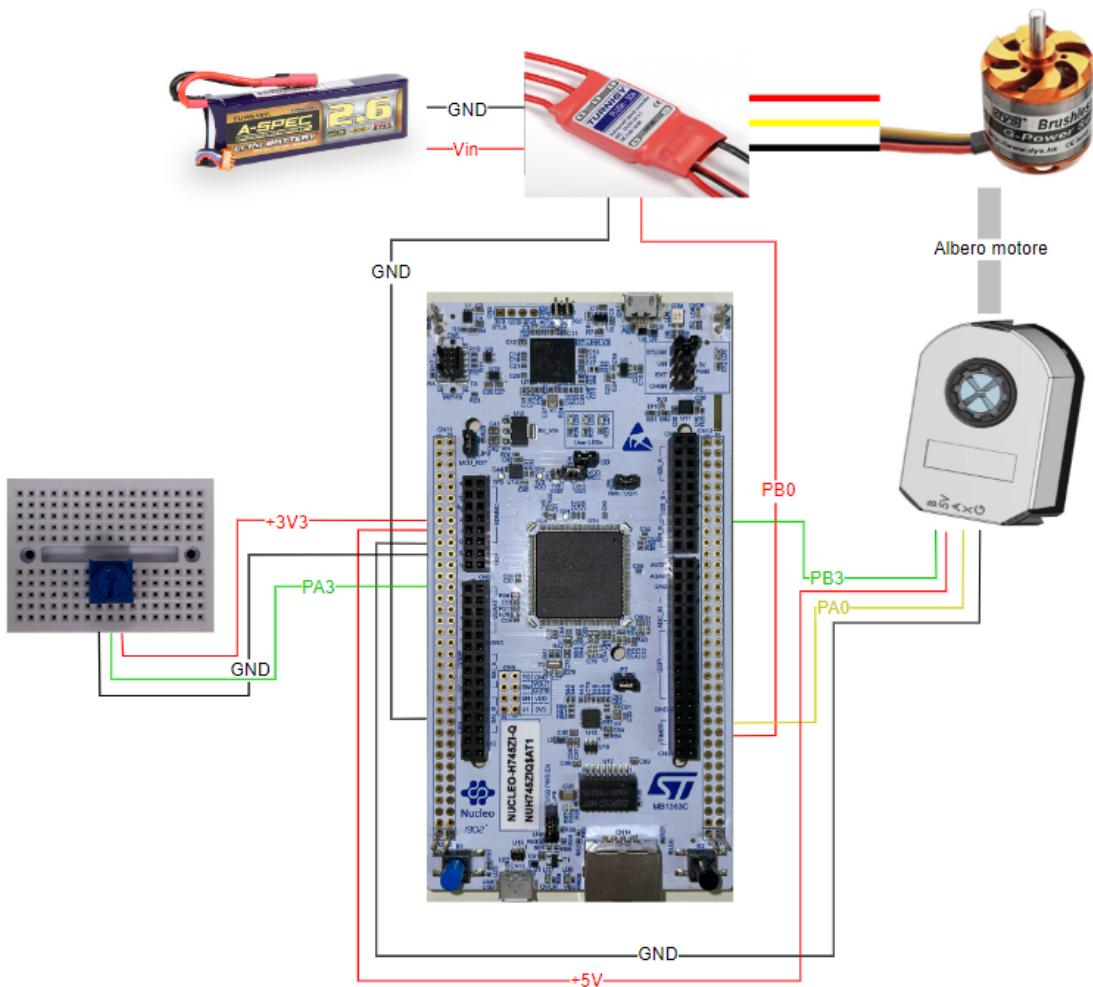


Figure 2.13: Schema dei collegamenti 1

La batteria e l' ESC sono collegati tramite i cavi dell'alimentazione; l'ESC e la scheda NUCLEO STM sono collegati tramite il circuito BEC; l'ESC e il motore brushless sono collegati tramite i cavi di alimentazione delle tre fasi; l'encoder è montato sull'albero motore; la scheda STM e l'encoder sono collegati, tramite una breadboard ai pin dell'encoder; sempre sulla breadboard è montato il trimmer, collegato ai pin della scheda.

2.8.2 Setup banco

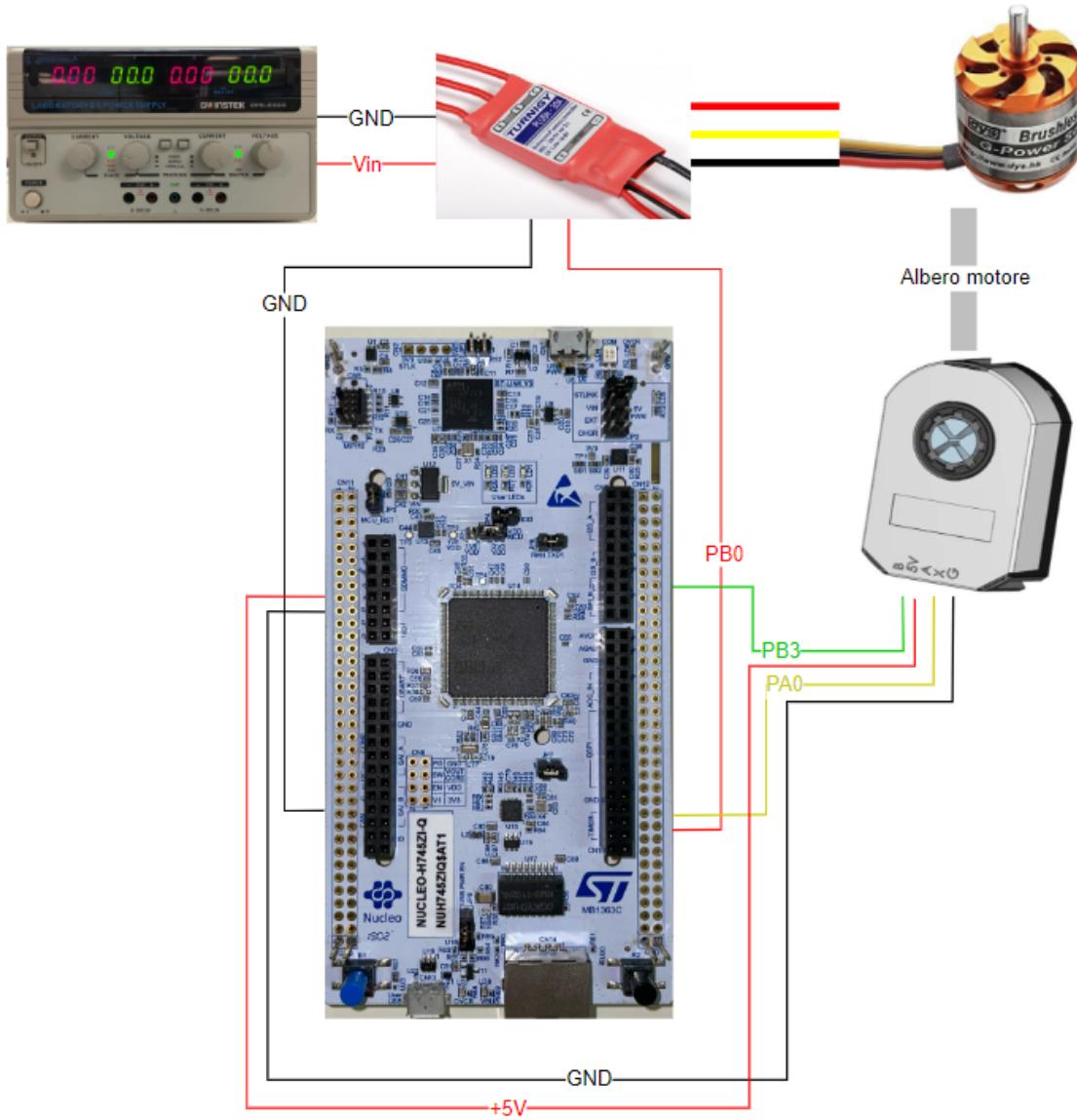


Figure 2.14: Schema dei collegamenti 2

In figura 2.14 è mostrato un secondo schema dei collegamenti in cui la sorgente di tensione è il banco di alimentazione, che permette di mantenere la tensione in ingresso stabile ad un valore scelto da noi. Inoltre abbiamo rimosso il trimmer e sostituito con un codice in grado di generare un duty incrementale.

3 Software

In questo capitolo vengono descritti brevemente gli strumenti software utilizzati in fase di test. Nello specifico vengono presentati sia STM32CubeIDE (versione 1.13.2), l'ambiente di sviluppo utilizzato per la programmazione della STM32, sia gli strumenti utilizzati per l'acquisizione dei dati tramite seriale (Serial Tools) e (Putty).



Figure 3.1: *STM32CubeIDE*

3.1 STM32CubeIDE

STM32CubeIDE è una piattaforma di sviluppo C/C++ avanzata con configurazione periferica, generazione di codice, compilazione di codice e funzionalità di debug per microcontrollori e microprocessori STM32. Si basa sul ®framework Eclipse /CDTTM e sulla toolchain GCC per lo sviluppo e GDB per il debug. Permette l'integrazione delle centinaia di plugin esistenti che completano le funzionalità dell'Eclipse - IDE.

STM32CubeIDE integra le funzionalità di configurazione e creazione di progetti STM32 di STM32CubeMX per offrire un'esperienza all-in-one degli strumenti e risparmiare tempo di installazione e sviluppo. Dopo la selezione di un MCU STM32 o MPU vuoto, o microcontrollore o microprocessore preconfigurato dalla selezione di una scheda o la selezione di un esempio, viene creato il progetto e generato codice di inizializzazione. In qualsiasi momento durante lo sviluppo, l'utente può tornare all'inizializzazione e alla configurazione delle periferiche o middleware e rigenerare il codice di inizializzazione senza alcun impatto sul codice utente.

STM32CubeIDE include analizzatori di build e stack che forniscono all'utente informazioni utili sullo stato del progetto e sui requisiti di memoria.

STM32CubeIDE include anche funzionalità di debug standard e avanzate, tra cui visualizzazioni dei registri core della CPU, memorie e registri periferici, nonché clock variabile dal vivo, interfaccia Serial Wire Viewer o analizzatore di guasto.

3.2 Putty

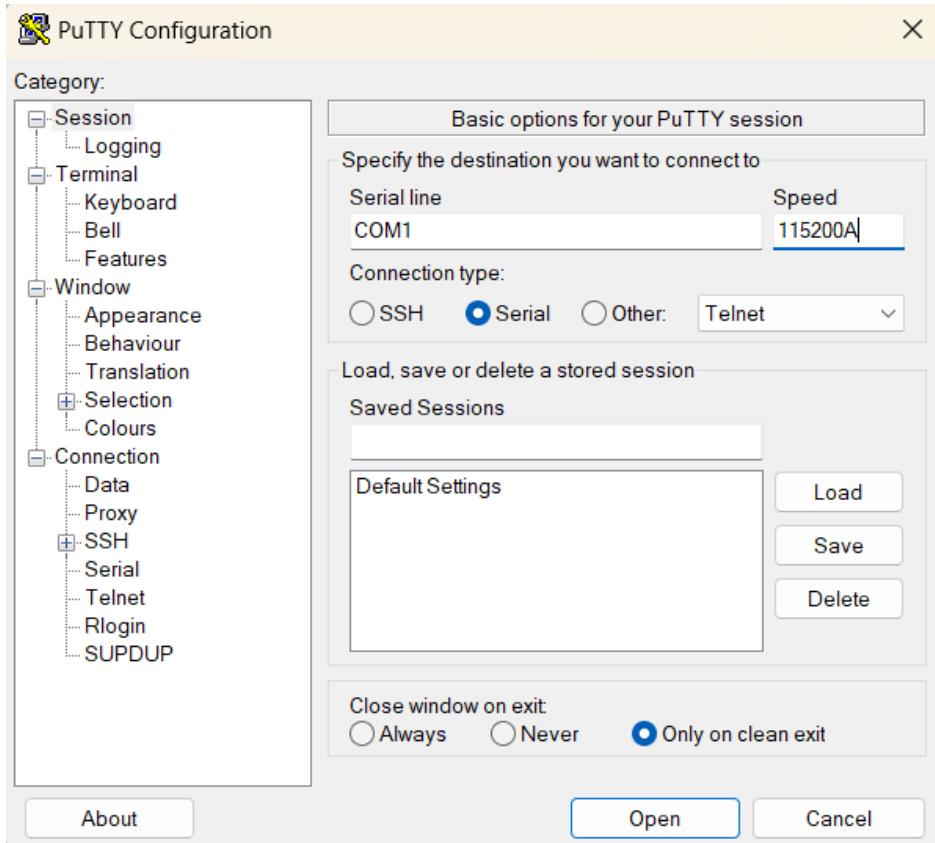


Figure 3.2: Putty

Putty è un client SSH che permette la visualizzazione in tempo reale dei dati provenienti dalla scheda STM.

3.3 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 3.2).

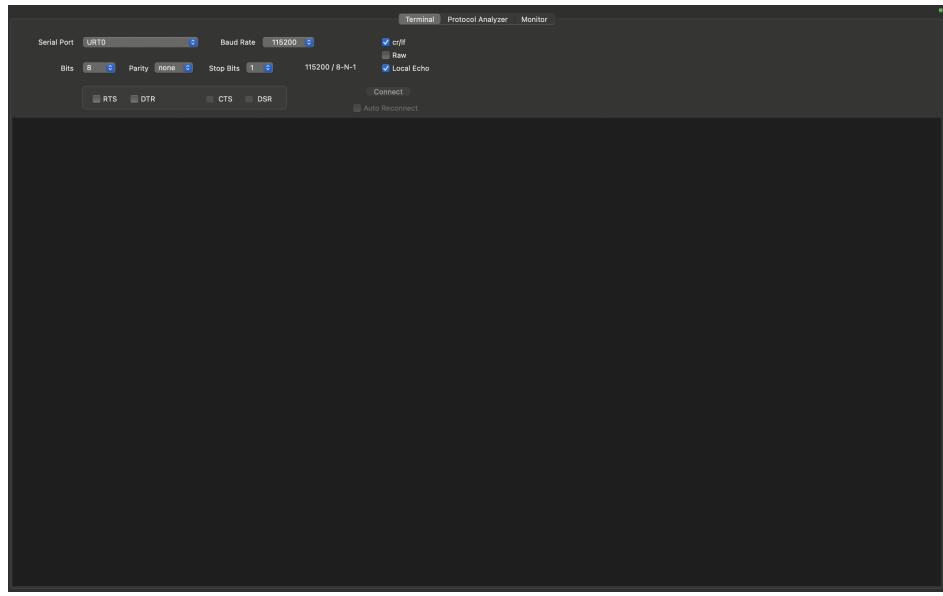
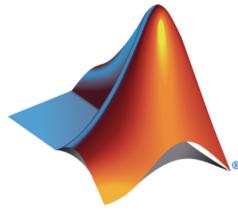


Figure 3.3: *Serial Tools*

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 3.3 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.

3.4 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.

Grazie a questo programma abbiamo potuto visualizzare i risultati dei nostri test tramite grafici.

3.5 Diagramma di flusso

In questo progetto abbiamo utilizzato due programmi separati per poter gestire diversamente i nostri componenti.

Il diagramma di flusso in figura 3.4 mostra il funzionamento del codice utilizzato nei primi test, che permette di variare il duty del motore tramite l'utilizzo di un trimmer.

Il diagramma è molto semplice, anzitutto vengono inizializzate le variabili periferiche, viene svolta la procedura di calibrazione del motore, all'interno del ciclo WHILE per ogni tempo di clock viene effettuata la lettura dei dati del trimmer che modifica il duty cycle e di conseguenza viene pilotato il motore, infine ogni 4 ms viene svolta la lettura dell'encoder e la stampa dei dati raccolti.

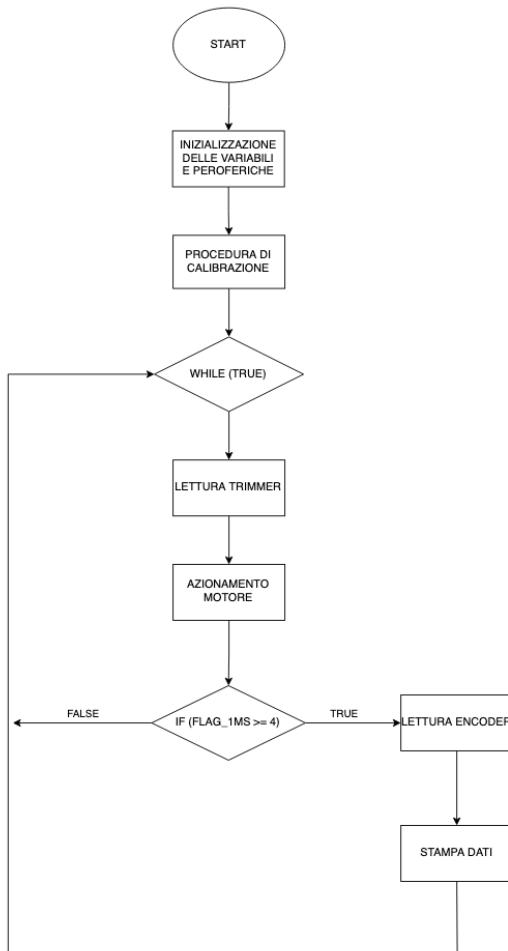


Figure 3.4: Diagramma di flusso del codice sviluppato

Il diagramma di flusso in figura 3.5 mostra il funzionamento del codice utilizzato per poter variare il duty del motore in modo incrementale.

In questo caso viene sostituito il controllo del duty cycle, precedentemente effettuato tramite trimmer, con un controllo gestito tramite codice incrementale. L'avvio dell'incremento del duty è controllato dallo user button tramite una flag.

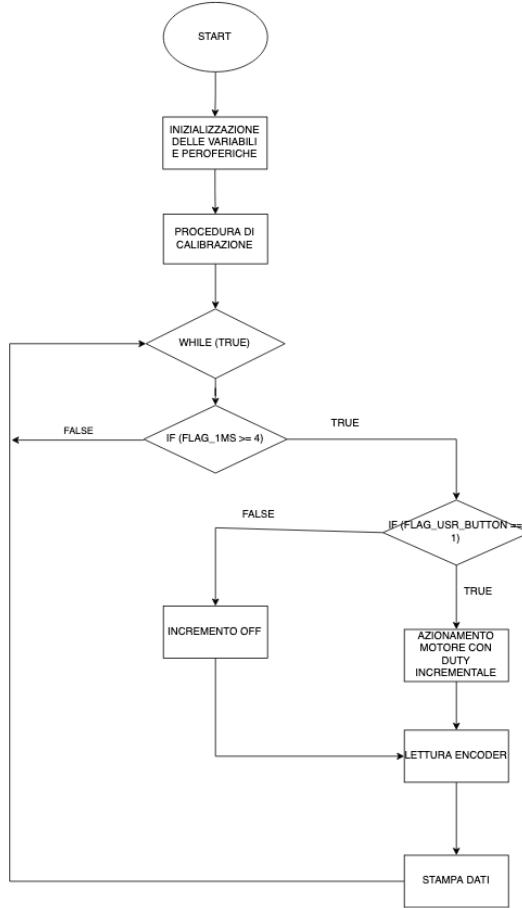


Figure 3.5: diagramma di flusso 2

3.6 Gestione singoli componenti

Di seguito riportiamo la configurazione del tempo di clock, dato che rimane invariata per tutti i componenti. Il tempo di clock è stato impostato a 400Hz perché abbiamo riscontrato problemi con un valore di 480Hz, questo comunque non compromette i risultati in quanto non abbiamo bisogno di prestazioni elevate.

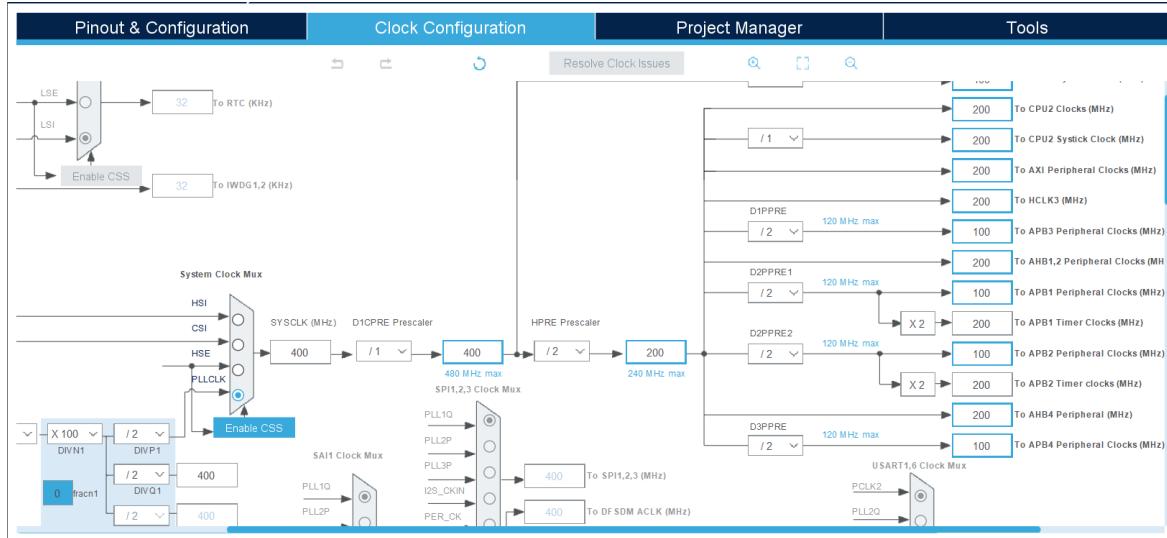


Figure 3.6: Tempo di clock

3.6.1 Gestione calibrazione ESC

Nell'uso comune l'ESC andrebbe calibrato tramite il radio comando, infatti secondo il manuale per eseguire la calibrazione bisogna prima portare l'acceleratore al massimo, aspettare almeno due secondi e portarlo nuovamente al minimo. Non disponendo del radio comando abbiamo eseguito la procedura codice utilizzando un timer per inviare un segnale PWM.

Anzitutto impostiamo il timer, in questo caso TIM3 in PWM generation mode e lo assegnamo al Cortex-M4. Abbiamo scelto il channel 3 che corrisponde al pin PB0, nel quale risiede il collegamento tra la scheda e l'ESC.

The screenshot shows the ESC2.0.ioc software interface. The main title bar reads "ESC2.0.ioc - Pinout & Configuration". Below it, there are two tabs: "Pinout & Configuration" and "Clock Configuration". The "Clock Configuration" tab is active, showing a sub-section titled "Software Packs" which is expanded to show "TIM3 Mode and Configuration".

In the "Mode" section, there is a table for "Runtime contexts" with columns for Cortex-M7, Cortex-M4, and PowerDomain. The "Cortex-M4" column has a checked checkbox under "PowerDomain", and the value "D2" is highlighted in blue. Below this table are several dropdown menus and checkboxes for configuring Slave Mode, Trigger Source, Clock Source, Channel1, Channel2, Channel3, Channel4, Combined Channels, ETR IO as Clearing Source, XOR activation, and One Pulse Mode.

On the left side of the interface, there is a search bar and a table listing various timer components: LPTIM2, LPTIM3, LPTIM4, LPTIM5, RTC, TIM1, TIM2, TIM3, TIM4, TIM5, TIM6, TIM7, TIM8, TIM12, TIM13, TIM14, TIM15, TIM16, and TIM17. The row for "TIM3" is selected, indicated by a blue background and checked checkboxes in the "Cortex-M4" and "PowerDomain" columns of the runtime context table.

Figure 3.7: PWM generation mode

Configuriamo Counter Clock TIM3 in modo che l' Update Event (UEV) frequency sia di 20 ms.

Dalla teoria si ha che:

$$UpdateEvent = \frac{TimerClock}{(Prescaler+1) \times (Period+1)}$$

Nel nostro caso il prescaler è uguale a 200-1, mentre il counter period 20000-1.

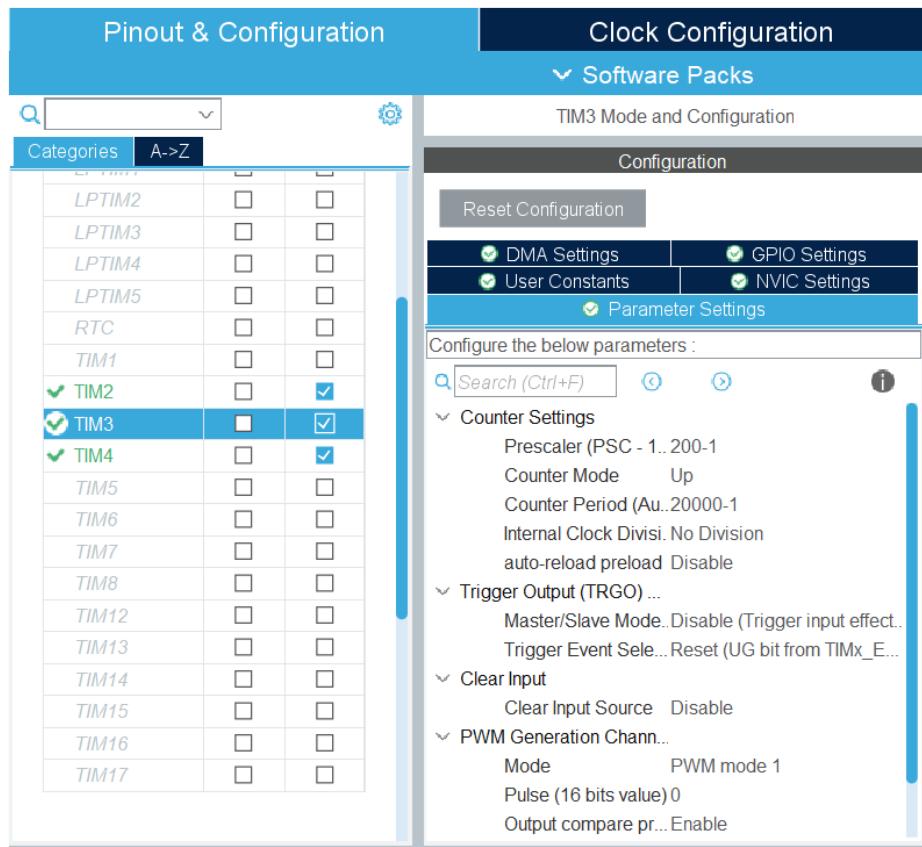


Figure 3.8: *TIM3 parameter settings*

Con questo codice simuliamo la procedura di calibrazione che andrebbe eseguita manualmente.

I valori di max_duty e min_duty vengono definiti nelle costanti, i quali valori vengono scelti arbitrariamente per effettuare i test e studiarne i limiti, questo argomento è approfondito nella sezione dei test [4 Test e risultati].

Il comando HAL_Delay(2000) ci assicura che la procedura venga eseguita correttamente, aspettando i due secondi necessari per la calibrazione.

Con il comando TIM3->CCR3 accediamo al duty register, con TIM3->ARR accediamo al valore massimo del registro. Questa procedura ci permette di far assumere al registro CCR3 prima il valore massimo e dopo due secondi il valore minimo del duty. Si ha infatti che il duty dipende direttamente dal valore del registro:

$$\delta = \frac{\text{MaxCNT} - \text{CCRx}}{\text{MaxCNT}}$$

```

1 // CODICE CALIBRATURA DUTY
2 TIM3->CCR3 = (max_duty/100)*TIM3->ARR;
3 HAL_Delay(2000);
4 TIM3->CCR3 = (min_duty/100)*TIM3->ARR;
5 HAL_Delay(2000);

```

3.6.2 Gestione Trimmer

I trimmer sono dei componenti elettrici costruiti in maniera tale da permettere di variare la loro resistenza. Nei nostri test ci siamo serviti di un trimmer per variare manualmente il duty.

Per utilizzare il trimmer assegnamo l' ADC1 al Cortex-M4 e abilitiamo il canale di input IN15. IN15 corrisponde al pin PA3. Tramite l' ADC (analogue digital converter) possiamo convertire la differenza di tensione registrata ai capi del trimmer in un valore numerico digitale.

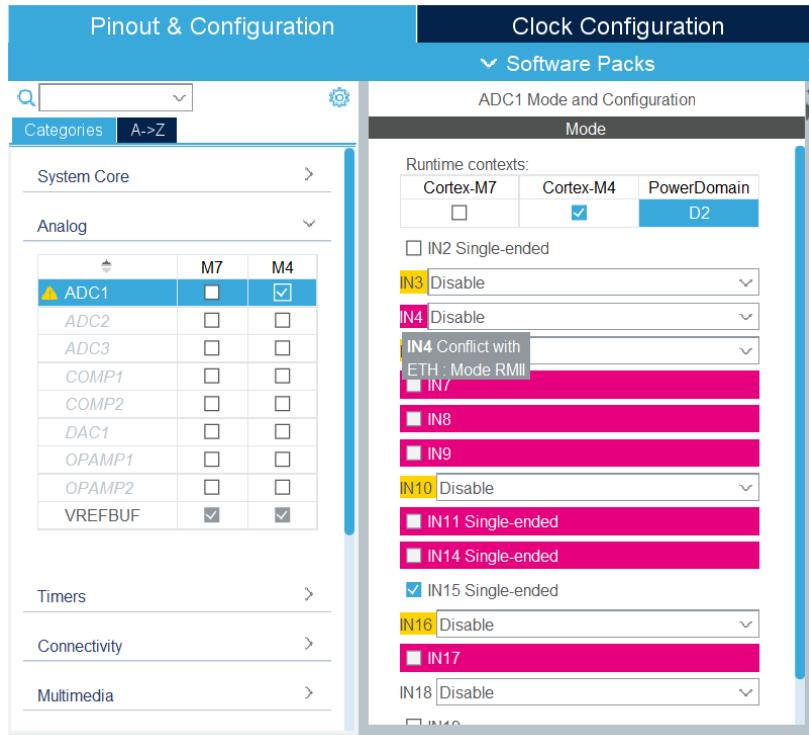


Figure 3.9: ADC1

Nei Parameter settings impostiamo la Discontinuous conversion mode.

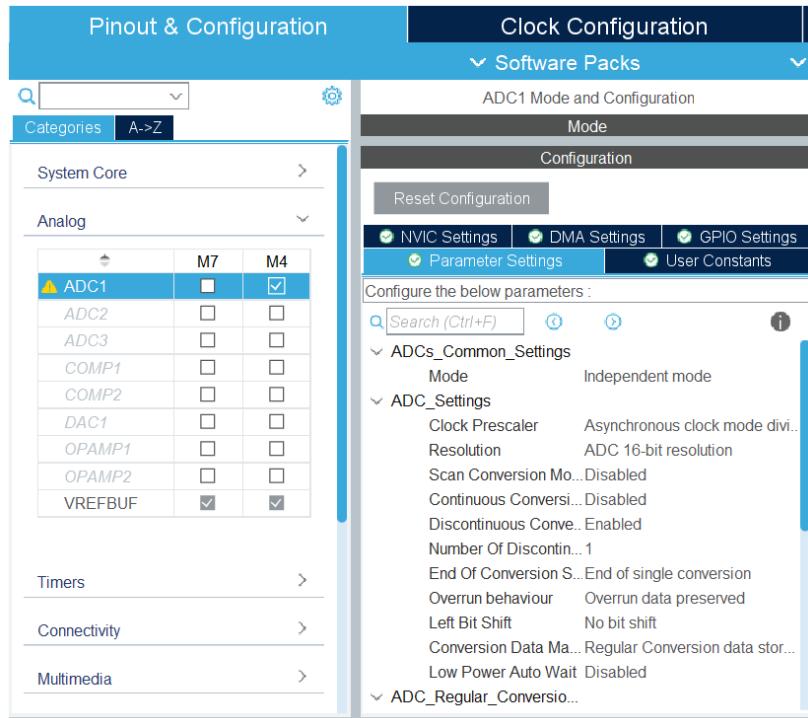


Figure 3.10: ADC1 Parameter settings

Nel codice, innanzitutto definiamo le variabili: timeout è un limite di tempo in ms impostato per cui, se il programma non ottiene informazioni entro quel tempo, prosegue il codice indipendentemente; CountValue è una variabile binaria che deve essere convertita dall' ADC, Volt è il valore decimale che si ottiene dalla conversione del valore del registro in tensione misurata.

```
1 int timeout=50;
2 uint32_t CountValue;
```

Viene mostrato il codice che gestisce l'acquisizione dei dati, CountValue assumerà il valore del registro ADC1, Volt assumerà il valore di CountValue convertito in decimale, infine la variabile duty viene definita in modo da variare tra il min_duty e il max_duty

```
1 //LETTURA DUTY TRIMMER
2 CountValue = HAL_ADC_GetValue (&hadc1);
3 Volt = CountValue * 3.3 / (65536-1);
4 float duty = ((Volt/3.3) * ((max_duty - min_duty)/100)) * min_duty/100);
```

3.6.3 Gestione Encoder

Per utilizzare l'encoder impostiamo il TIM2 in Encoder Mode.

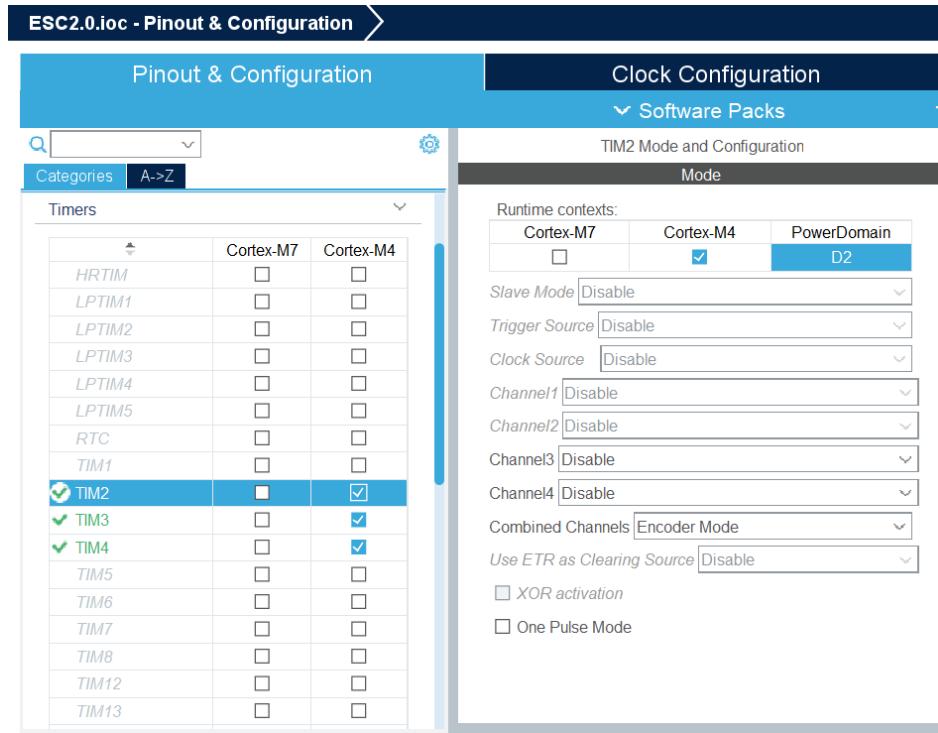


Figure 3.11: Timer in Encoder mode

Attivando i canali TI1 e TI2, sfruttiamo i due canali di quadratura dell' encoder stabilendo K = 4.

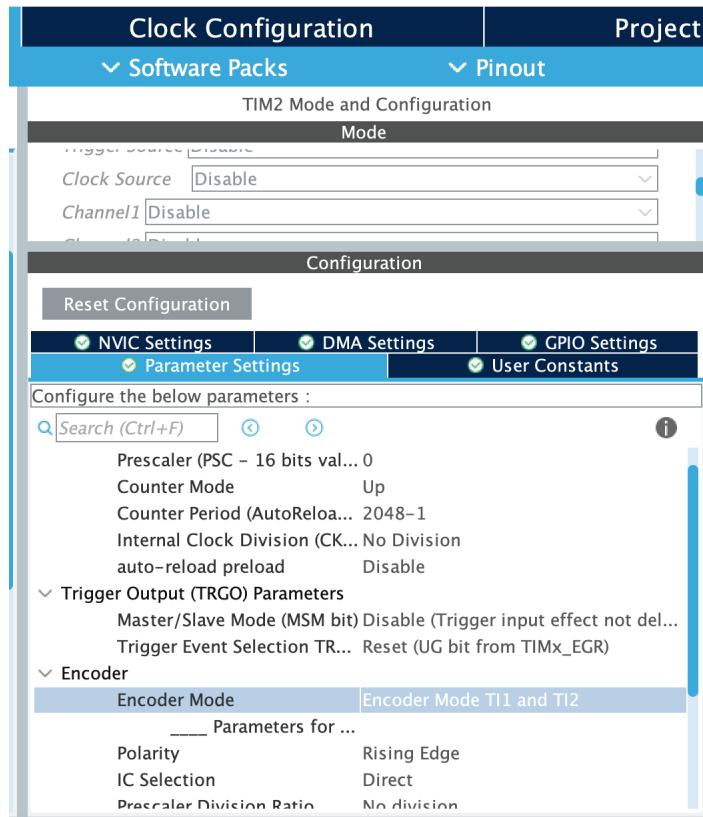


Figure 3.12: Parametri TIM2

Come mostrato in figura 2.10, il nostro encoder deve essere impostato con un PPR di 512 per avere la risoluzione necessaria a misurare correttamente la velocità del motore.

Di seguito riportiamo il codice usato per ottenere i dati dall'encoder.

Sono state definite le costanti: ENCODER_PPR = 512, rappresenta la risoluzione dell'encoder; ENCODER_COUNTING_MODE = 4; GEARBOX_RATIO = 1. Quest'ultimo parametro è stato settato ad 1 in quanto non abbiamo un riduttore di velocità, poiché le misurazioni vengono eseguite sull'albero motore. Nel caso in cui venisse letto un valore da un albero connesso dopo un riduttore, il GEARBOX_RATIO deve valere il rapporto del riduttore. La lettura dell'encoder viene effettuata ogni 4ms, utilizzando il TIM4, introdotto nella prossima sezione.

```

1 //LETTURA ENCODER
2 counts = TIM2->CNT;
3 delta_angle = (double)counts * 360 / (ENCODER_PPR * ENCODER_COUNTING_MODE
4 * GEARBOX_RATIO);
5 diff_angle = delta_angle - old_delta_angle;
6 dir = (TIM2->CR1 & 0X0010)>>4;
7 if(dir == old_dir){
8     if(dir == 0 && diff_angle < 0){
9         diff_angle = 360 + diff_angle;
10    } else if (dir == 1 && diff_angle > 0){
11        diff_angle = diff_angle - 360;
12    }
13}
```

```

11     }
12 }
13 speed = diff_angle/dt;
14 speed_RPM = speed*60/360; //60 secondi, 360 gradi.
15
16 old_delta_angle = delta_angle;
17 old_dir = dir;
18
19 //STAMPA SU PUTTY IL DUTY E LA VELOCITA'
20 printf("%f;%f\r\n", duty, speed_RPM);

```

3.6.4 Gestione duty incrementale

Per effettuare alcuni test è stato utile sviluppare un codice che generasse un duty incrementale. Impostiamo quindi il TIM4 per generare una flag ogni 1ms (UEV frequency = 1ms)

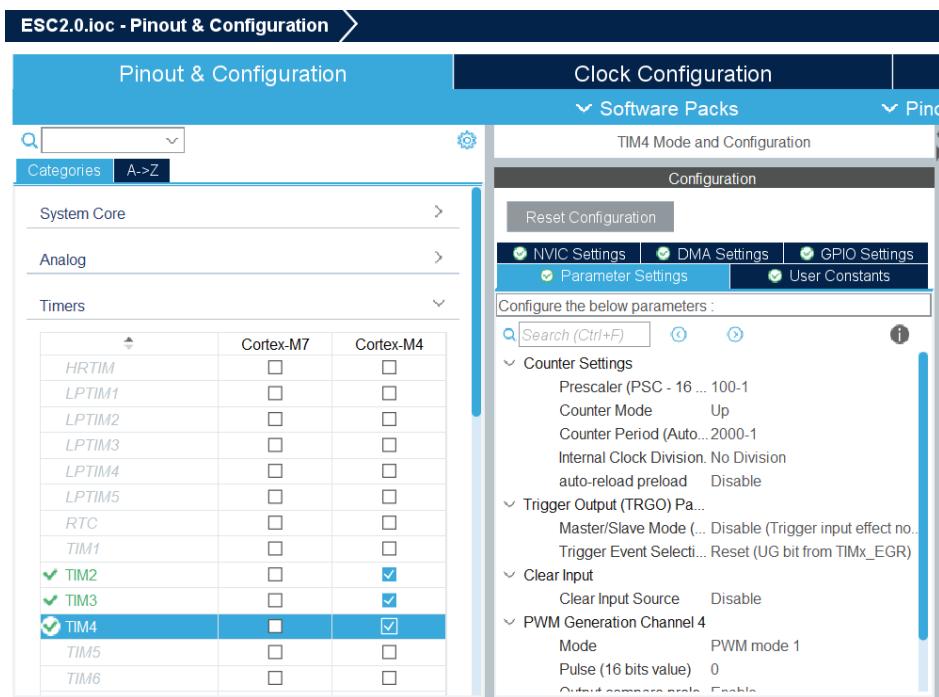


Figure 3.13: TIM4 Parameter Settings

Di seguito è riportato il codice per attivare la flag di 1ms:

```

1 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim){
2     if(htim == &htim4){
3         flag_1ms = flah_1ms + 1;
4     }
5 }

```

Usiamo inoltre l'user button per attivare o disattivare quando necessario l'incremento. Al corrispondente PIN PC13 abilitiamo la linea EXTI13, impostiamo la GPIO mode in external

interrupt mode with Rising edge.

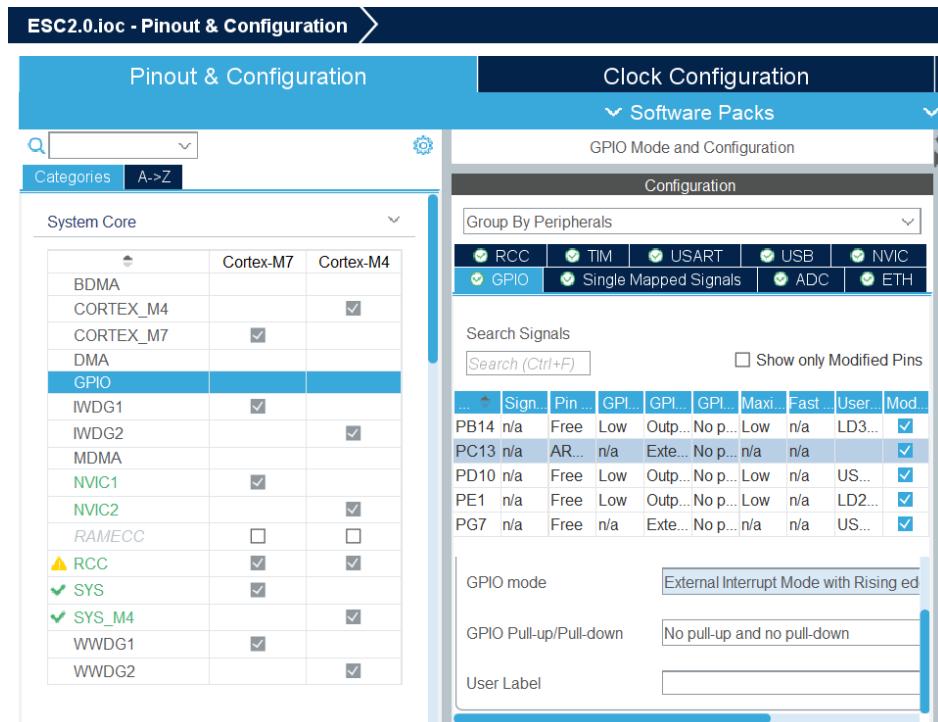


Figure 3.14: *USRButton Interrupt*

```

1 // FUNZIONE FLAG PER ENCODER INCREMENTALE
2 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin){
3
4     if(GPIO_Pin == GPIO_PIN_13){
5         if(flag_bottone == 0){
6             flag_bottone = 1;
7         }
8         else
9             flag_bottone = 0;
10    }
11 }
```

Il codice è stato pensato per incrementare il duty dal duty_min al duty_max per poi diminuirlo nuovamente in modo periodico.

```

1 // CODICE PER DUTY INCREMENTALE
2 if (flag_bottone == 1){
3     if (flag_incremento == 1){
4         duty_inc = duty_inc + inc; // inc = 0.00001
5         TIM3->CCR3=TIM3->ARR*duty_inc;
6         if (duty_inc > max_duty/100)
7             flag_incremento = 0;
8
9     }
10 }
```

```

11     duty_inc = duty_inc - 0.00001;
12     TIM3->CCR3=TIM3->ARR*duty_inc;
13     if (duty_inc < min_duty/100)
14         flag_incremto = 1;
15     }
16 }
```



```

1 // CODICE PER LA TRASMISSIONE DEI DATI TRAMITE USART
2 int __io_putchar(int ch){
3     HAL_UART_Transmit(&huart3, (uint8_t*)&ch, 1, 0xFFFF);
4     return ch;
5 }
```

Il codice riportato qui sopra ha lo scopo di trasmettere i dati tramite USART, nello specifico andiamo ad usare la funzione HAL_UART_Transmit che trasmetterà ogni singolo byte(char) tramite ST-LINK USART3 precedentemente abilitato nell'IOC.

3.7 Funzionamento complessivo

3.8 Codice Matlab

In questo capitolo spieghiamo brevemente il funzionamento del codice Matlab usato.

Il codice consente di acquisire i dati provenienti da SerialTools su MatLab. I dati di nostro interesse sono il duty e la velocità misurata, convertita in RPM. Tc è il tempo di campionamento, il cui valore dipende da come è stato impostato il relativo timer della scheda STM. Infine, ci permette di graficare i dati registrati tramite la funzione tiledlayout.

```

1 clearvars;
2 close all;
3 clc;
4
5 %% PARAMETRI ACQUISIZIONE
6 port = '/dev/tty.usbmodem142403';
7 Baund_Rate = 115200.0;
8 Data_Bits = 8;
9 Stop_Bits = 1;
10
11 if ~isempty( instrfind )
12     fclose( instrfind );
13     delete( instrfind );
14 end
15 s = serialport( port, Baund_Rate, 'DataBits', Data_Bits, ...
16     'StopBits', Stop_Bits ); %collegamento alla porta collegata
17     alla scheda
18 while ( isempty( seriallist ) ) % attende che la porta sia
19     disponibile
20 end
```

```

19 %% ACQUISIZIONE DATI
20 Tc=0.004;
21 duty = zeros(1000,1); %definizione matrice per valori del duty
22 speed = zeros(1000,1); %definizione matrice per valori della
    'velocita'
23 ValSpeed = zeros(1000,1);
24 i=1;
25 while true
26     try
27         %FORMAT: DUTY, SPEED
28         S = readline(s); % lettura riga
29         A = str2num(S); %conversione da stringa in numero
30         duty(i)=A(1);
31         speed(i)=A(2);
32         i=i+1;
33     catch
34         if (~isempty(S)) % controllo se il dato e' stato
acquisito
35             fprintf(S)
36         else
37             fprintf("Non acquisito.\n");
38             break
39         end
40     end
41 end
42 %%GRAFICI
43 figure;
44 tiledlayout(2,1); % comando per mostrare a schermo entrambi i
    grafici contemporaneamente
45 nexttile
46 plot((1:length(duty))*Tc,duty) % comando grafico del duty
47     ylabel("Duty [%]");
48     xlabel("Time [s]");
49 nexttile
50 plot((1:length(speed))*Tc,speed) % comando grafico della
    'velocita'
51     ylabel("Speed [RPM]");
52     xlabel("Time [s]");
53 %% Chiusura porta
54 fclose (s);
55 delete (s);

```

4 Test e risultati

4.1 Test con trimmer duty

In questi test, effettuati semplicemente impostando valori del duty cycle sul codice e con l'ausilio di un trimmer, abbiamo cercato risposte sui limiti di impostazione del duty per la fase di calibrazione dell'ESC. Nonostante sul manuale non sia specificato nulla riguardo limiti del range ammissibile, non tutti i valori consentono un corretto utilizzo. Durante la prima fase di studio del materiale fornito e testando il codice, abbiamo riscontrato anomalie coerenti con i risultati ottenuti precedentemente dai nostri colleghi. La calibrazione del motore non avveniva correttamente in range del duty non comune, per range più ampi infatti, come ad esempio 5-30% o 5-50%, l'ESC non emetteva i suoni che confermano la corretta calibrazione. È stato possibile aggirare il problema aumentando manualmente il valore del duty minimo impostato nel codice durante la fase di calibrazione (vedi [2.4.2] Procedura di calibrazione). Le cause di questo comportamento, nonostante i numerosi test effettuati su range differenti, rimangono incomprese.

Tuttavia facciamo presente che le problematiche riscontrate non avvengono nel range tra il 5-15% e che questi valori sono spesso utilizzati nel controllo di droni e veivoli di questo genere, anche i radiocomandi utilizzati per il pilotaggio non sforano lo standard. Questo ci suggerisce che, nonostante possa essere più comodo utilizzare range più ampi per una precisione maggiore, l'ESC abbia dei limiti invalicabili, anche se non specificati; riteniamo perciò non profigue ulteriori ricerche.

4.2 Test con banco di alimentazione duty/velocità

In questi test tramite un encoder abbiamo visualizzato il rapporto tra la velocità del motore brushless e il duty. Il nostro obiettivo era riuscire a dare robustezza ai risultati ottenuti e individuare i valori più adatti al controllo del motore. Nei test abbiamo utilizzato valori standard del duty minimo e massimo come 5-10%, 8-12%, 5-15%. Abbiamo svolto questi test con una tensione sempre pari a 15V aspettandoci quindi una velocità massima di $V_{max} = Kv \times 15$, che con un Kv di 910 RPM/V del nostro motore è pari a $V_{max} = 13650$

4.2.1 5-10%

Condizioni del test:

- Tensione = 15V
- Duty = 5-10%

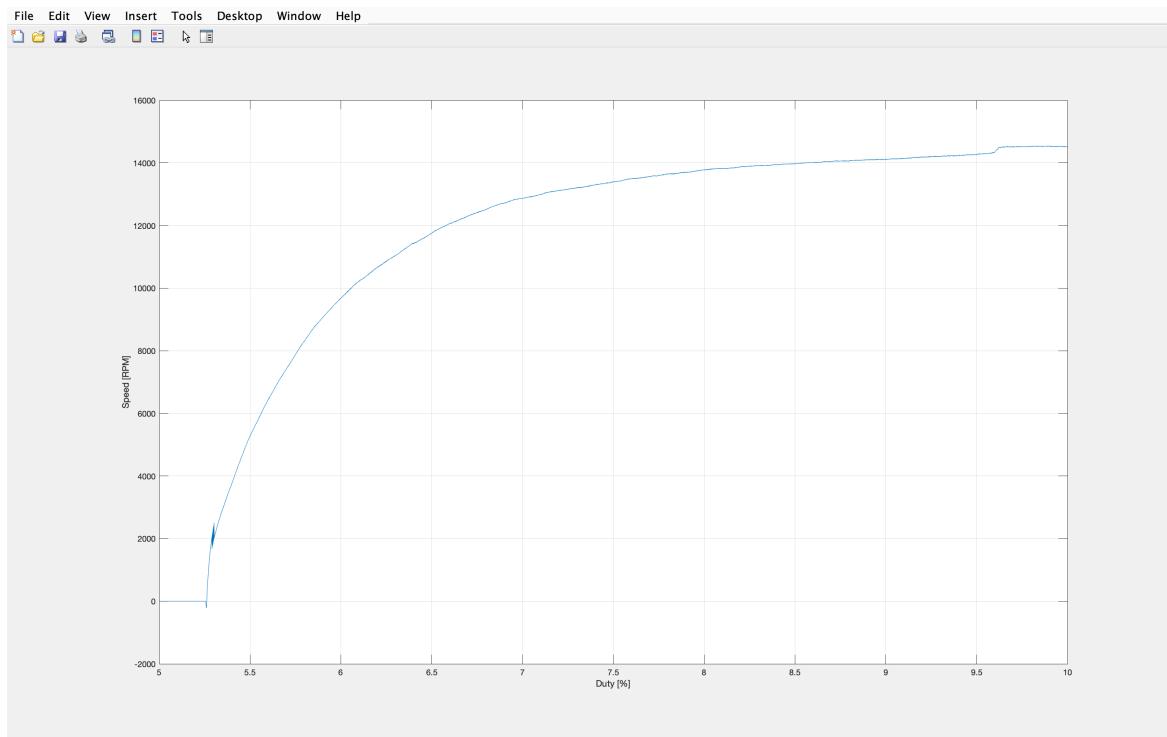


Figure 4.1: grafico duty 5-10

Il grafico 4.1 ci mostra l'andamento della velocità in RPM rispetto al duty impostato. La velocità raggiunta è leggermente superiore alla stima precedente. La velocità cresce velocemente nel primo range di duty (5-7%) e si stabilizza nel tratto finale (8-10%). I risultati ottenuti sono conformi alle nostre aspettative, tuttavia ci siamo accorti di un comportamento non mostrato dai vecchi test: raggiunto un valore di duty pari al 95% del duty max il motore comincia a girare ad una velocità maggiore pressochè costante.

4.2.2 8-12%

Condizioni del test:

- Tensione = 15V
- Duty = 8-12%

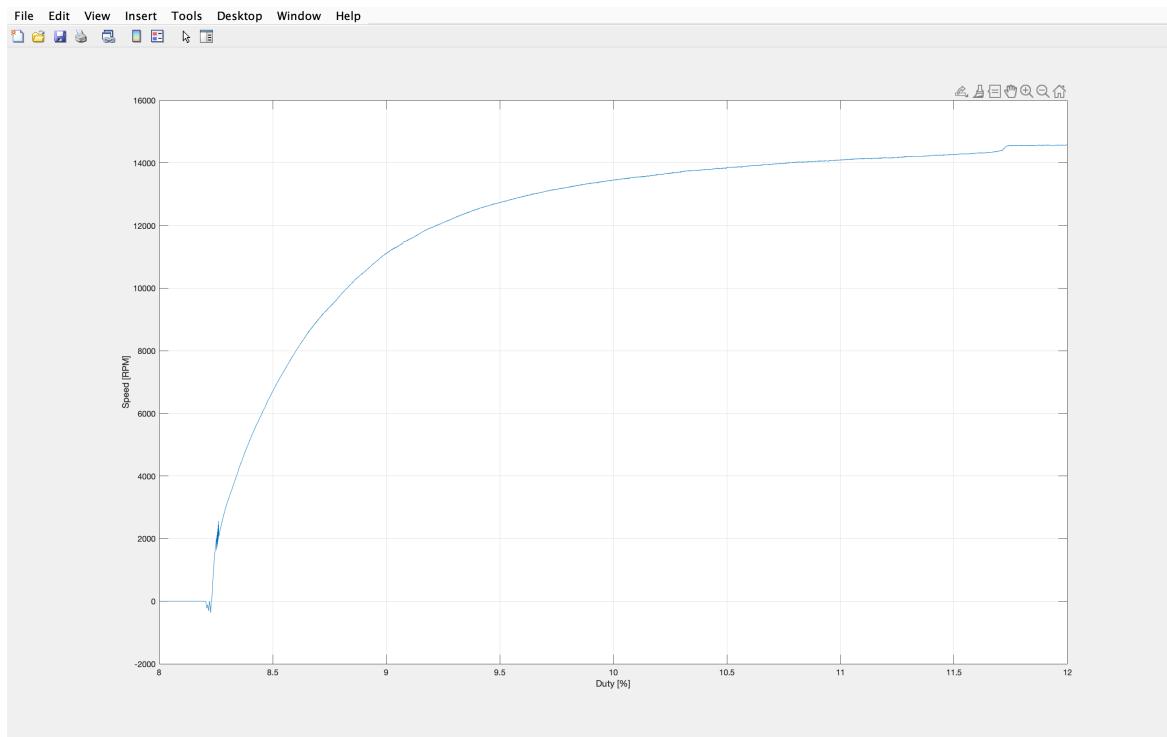


Figure 4.2: grafico duty 8-12

I risultati di questo test sono analoghi a quello precedente, mostrando come il valore del duty impostato non varia le prestazioni del motore offrendo quindi maggiore flessibilità alle esigenze. Anche in questo caso ad un valore di duty di circa 95% del duty max il motore passa velocemente ad una velocità più elevata.

4.2.3 5-15%

Condizioni del test:

- Tensione = 15V
- Duty = 5-15%

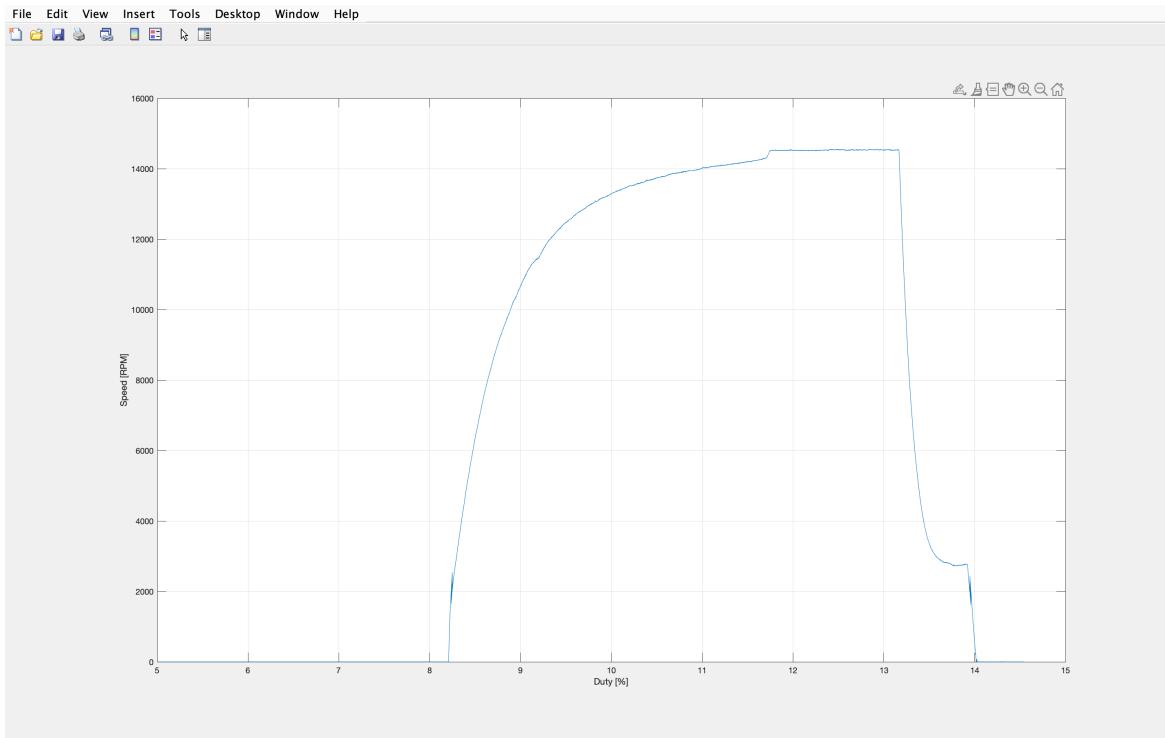


Figure 4.3: grafico duty 5-15

Questo test è risultato più critico rispetto ai precedenti. L'andamento della velocità è analogo ai precedenti, tuttavia come mostrato dal grafico 4.3 il motore raggiunto un duty di circa 13,1% subisce un brusco rallentamento, mantenendo una velocità bassa(2500-3000 RPM), finchè con un duty sempre pari al 14% si ferma. Questo comportamento viene affrontato nella sezione 4.4 Una ulteriore irregolarità si presenta all'accensione, infatti il motore comincia a ruotare a valori molto lontani dal duty minimo impostato, comportamento riscontrato in maniera del tutto casuale e che necessita nuove investigazioni.

4.3 Test con banco di alimentazione velocità/tensione

In questi test abbiamo sostituito l'alimentazione dell'ESC, invece di utilizzare la batteria a 4 celle abbiamo collegato il dispositivo al banco di alimentazione(setup2), permettendoci in questo modo di monitorare non solo la tensione ma anche la corrente inviata.

L'obiettivo principale è quello di studiare il comportamento del motore e le prestazioni con tensioni sempre minori.

4.3.1 Tensione=16.8V

Per effettuare questo test è stata modificata la parte riguardante l'encoder nell'ioc, attivando solamente il canale TI1 in modo da utilizzare un solo canale dell'encoder, inoltre è stato modificato il valore della variabile K da 4 a 2. In questo modo non si incorre nell'overflow e l'encoder è capace di calcolare i giri al minuto effettuati dal motore , dato che quest'ultimo tocca i 16000 giri al minuto.

Condizioni del test:

- Tensione = 16.8V
- Duty = 5-10%

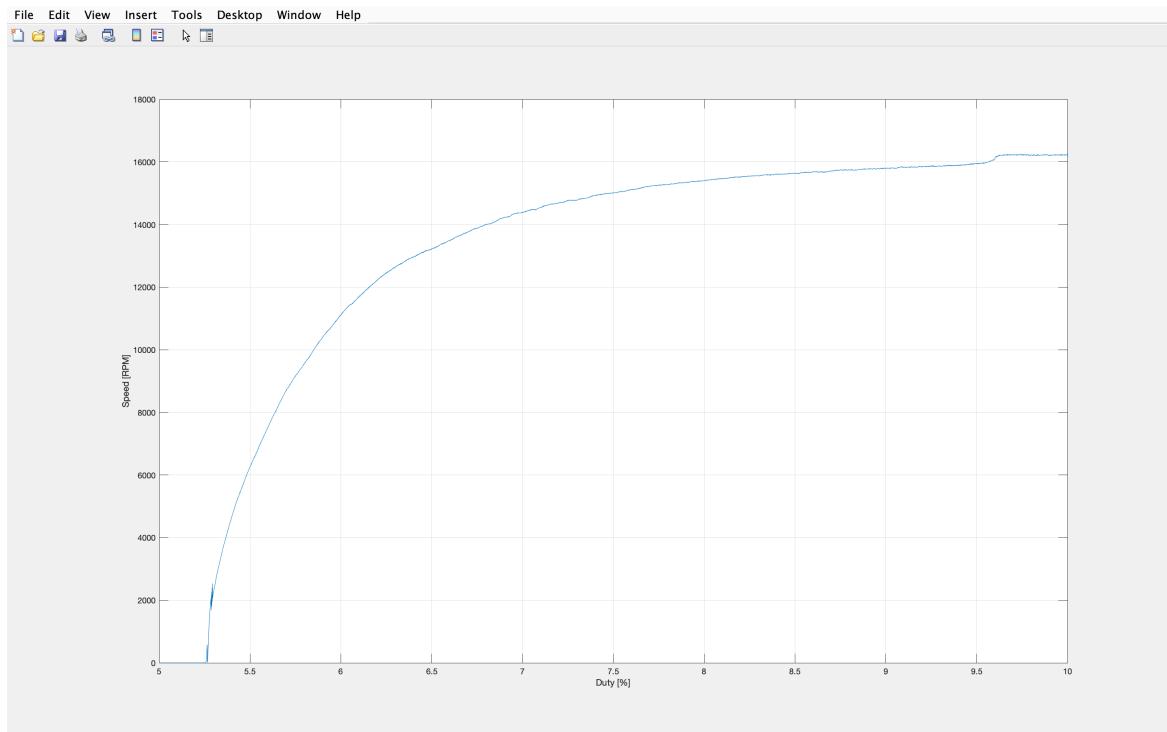


Figure 4.4: grafico tensione 16,8V

Abbiamo scelto la tensione per simulare la batteria completamente carica. Notiamo che la velocità massima raggiunta dal motore è maggiore di quella stimata (15288 RPM), di oltre 700 giri al minuto.

4.3.2 Tensione=15.0V

Condizioni del test:

- Tensione = 15V
- Duty = 5-10%

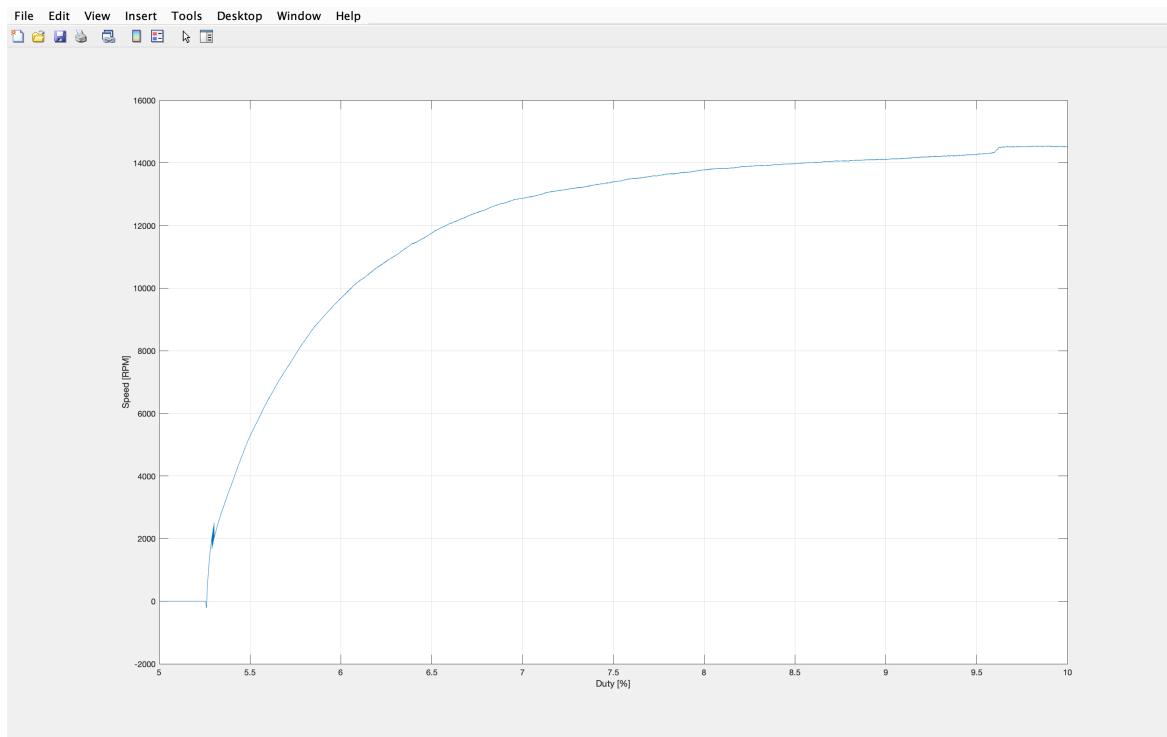


Figure 4.5: grafico tensione 15V

Diminuendo la tensione, simuliamo una batteria parzialmente scarica, il grafico mostra la riduzione delle prestazioni. Questo ci permette di valutare la conseguente variazione della spinta generata dal propulsore.

4.3.3 Tensione=13.0V

Condizioni del test:

- Tensione = 13V
- Duty = 5-10%

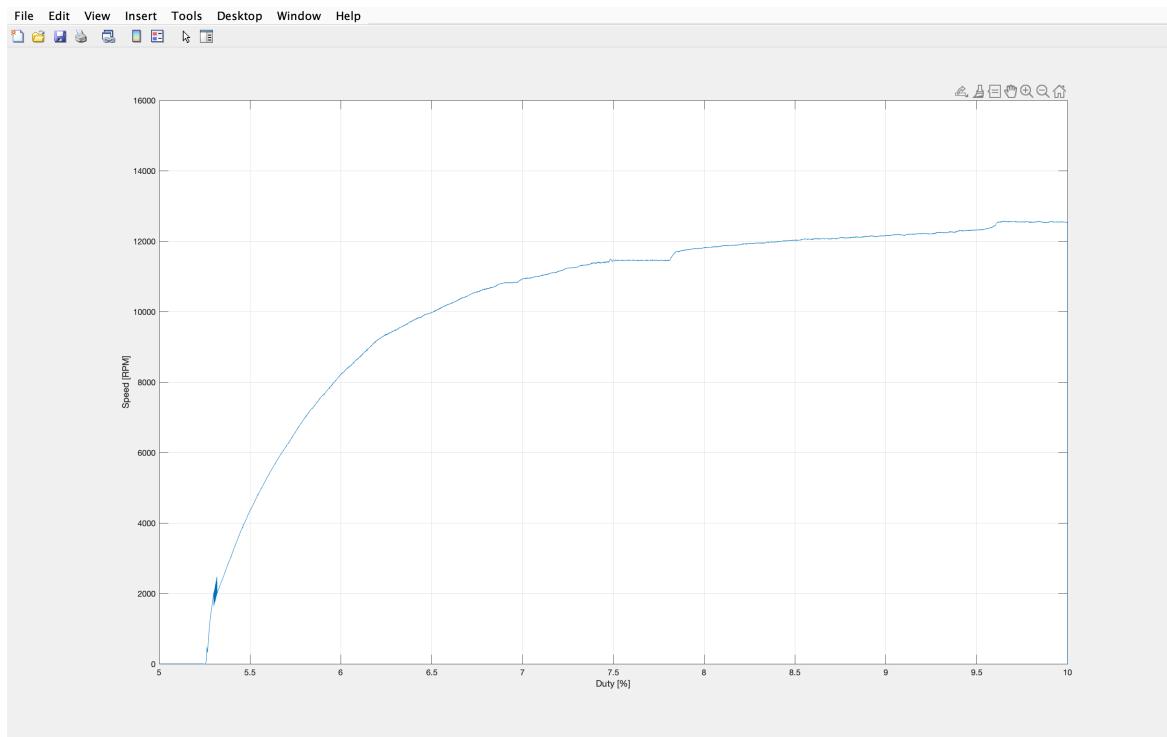


Figure 4.6: grafico tensione 13V

In questo test abbiamo simulato la tensione minima consentita dall'esc in modalità cut off alta (vedi 2.4.1). Notiamo che la misura della velocità è molto rumorosa, questo potrebbe essere causato dalle vibrazioni prodotte dal motore osservate durante questo test specifico.

4.4 Spegnimento del motore

Il secondo setup è stato anche utile per investigare sullo spegnimento inaspettato del motore. Impostando valori più alti di duty come 5-20% o 5-30%, una volta che il duty supera un valore pari a circa 13,7% si tiene ad una velocità bassa per due secondi per poi spegnersi ad un duty esattamente del 14%, il comportamento viene mostrato nei grafici MatLab 4.7 e 4.8. Questo avviene indipendentemente dal range impostato, il che ci fa intuire che il limite massimo di duty sia un valore fisso e non variabile.

Condizioni del test:

- Tensione = 15V
- Duty = 5-20%

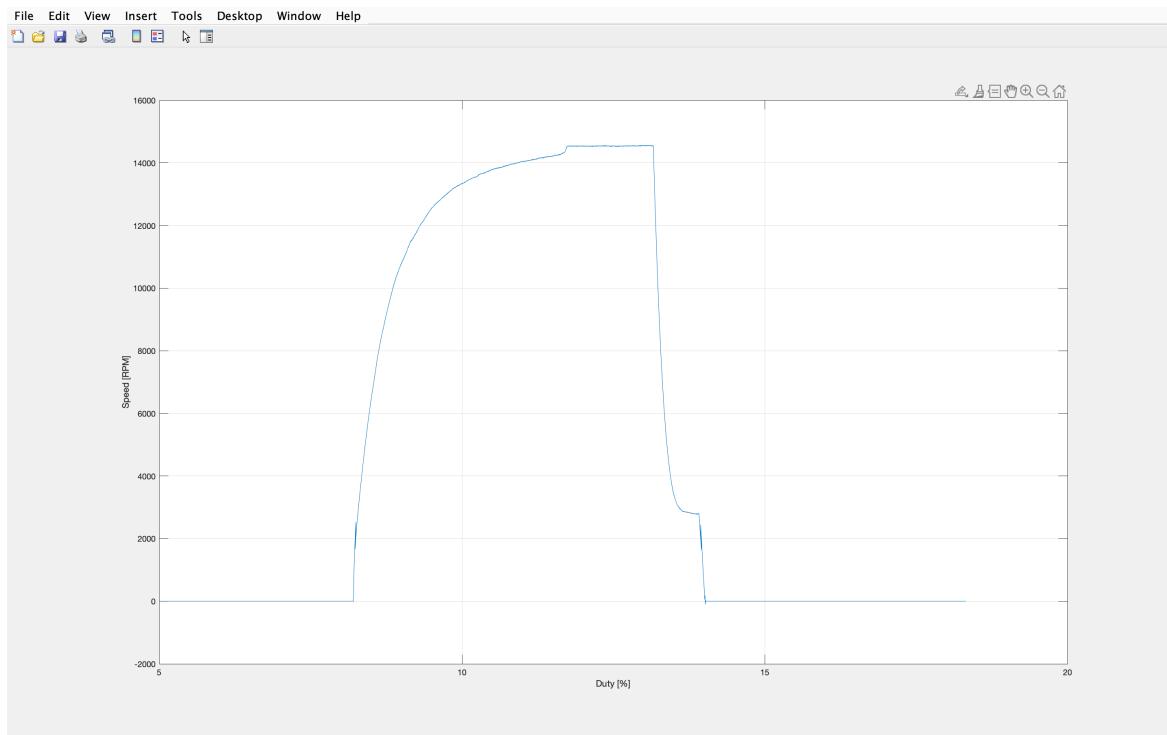


Figure 4.7: grafico duty 5-20

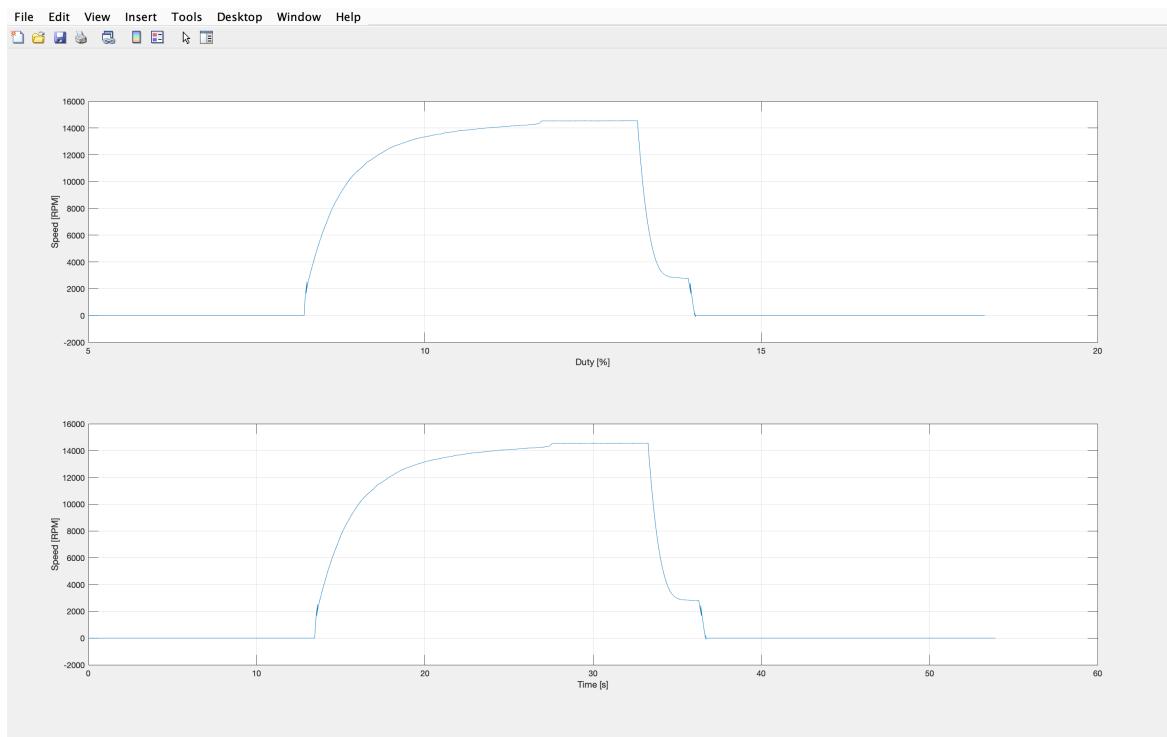


Figure 4.8: grafico relazione speed-duty and grafico relazione speed-tempo

4.4.1 Cut-Off

Inizialmente credevamo che valori troppo alti di duty richiedessero tensioni troppo alte non assicurate dalla batteria e che quindi portasse l'ESC ad entrare nella modalità di Cut-Off, impostata inizialmente su "Soft-cut" (vedi 2.4.1). Successivamente tramite il programmatore dell'ESC abbiamo impostato la modalità "Cut-Off" come mostrato in figura 4.9, la quale avrebbe dovuto far spegnere il motore immediatamente, tuttavia il comportamento è stato identico a quello precedente.



Figure 4.9: Programmatore in Cutoff

4.4.2 ESC Protection Mode

Il manuale dell'ESC non cita nessuna modalità di protezione che descrive il comportamento, perciò le cause del problema possono risalire o a un surriscaldamento dell'ESC (oltre i 110°), o una perdita del segnale, o collegamenti non affidabili [5]. Poichè riteniamo queste cause improbabili e inoltre il manuale non fa nessun riferimento ad uno spegnimento improvviso come descritto in precedenza, il problema potrebbe risiedere nel motore.

4.4.3 Tensioni e Correnti elevate per il motore

Non disponiamo del manuale ufficiale del motore ed è stato impossibile reperirlo, perciò abbiamo monitorato la tensione inviata e le correnti per individuare valori critici.

Collegando il banco di alimentazione all'ESC ci siamo assicurati che la tensione in ingresso non variasse; sempre tramite il banco e con l'ausilio del Multimetro ci siamo accorti che le correnti prima e durante lo spegnimento non variavano dai valori consentiti (1A, 1.5A). Un ultimo tentativo è stato quello di diminuire la tensione a valori come 12V o 10V, poichè la tensione nominale indicata nel datasheet del motore è di 15V, questo per poter escludere la possibilità

di tensioni troppo elevate sul motore, tuttavia quest'ultimo si comporta esattamente come in precedenza spegnendosi allo stesso modo una volta raggiunto il duty "massimo".

4.4.4 Conclusione

Con i nostri test abbiamo dimostrato che le cause dello spegnimento improvviso non sono connesse a:

- Tensioni troppo basse per l'ESC.
- ESC protection mode.
- Tensioni elevate per il motore.
- Correnti elevate per il motore.

In conclusione la causa del comportamento anomalo potrebbe risalire a valori non ammissibili di duty cycle o ad altre motivazioni a noi ignote e possibilmente citate nel manuale del motore brushless.

Conclusioni e sviluppi futuri

Uno dei compiti del nostro progetto era quello di porre il nostro sistema in diverse condizioni per identificare la migliore configurazione dell'ESC utile ai nostri colleghi al lavoro sulla stabilizzazione del drone. I primi test effettuati ci hanno confermato i risultati ottenuti in precedenza, tuttavia hanno portato alla luce un nuovo comportamento non menzionato nelle vecchie relazioni: la velocità del motore, prima di stabilizzarsi sul valore massimo, subisce un improvviso incremento negli ultimi valori del duty. Questo comportamento necessita ulteriori ricerche, non impone un limite al sistema e potrebbe trattarsi di una modalità di controllo del motore per aumentare i giri al minuto massimi.

Inoltre con i nostri test abbiamo evidenziato le differenti prestazioni al variare della tensione in ingresso, sottolineando l'importanza della carica della batteria.

Infine, uno dei nostri obiettivi era quello di scoprire le cause del comportamento anomalo del motore. Non siamo riusciti ad individuare con certezza il motivo, però tramite i test eseguiti abbiamo potuto escludere la maggior parte delle possibili cause, attribuendo l'origine del comportamento alla logica di funzionamento dell'ESC.

Tutti i nostri test seggeriscono che il duty abbia un limite massimo impostabile, questo non è descritto in nessun manuale da noi consultato, ma sfiorando il 14% di duty max il motore non si calibra correttamente, raggiunge la sua velocità massima, a prescindere dal range, poco prima del 14% per poi spegnersi esattamente a quel valore.

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.
- Approfondire le cause dei nuovi comportamenti anomali riscontrati durante i vari test.

- 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.
- 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.

Bibliografia

- [1] Lena Marison, "Tesi esc," 2022-2023. Relazione corso Laboratorio di Automazione.
- [2] G. INSTEK, "Bhgps-series." disponibile ONLINE: <https://www.gwinstek.com/en-global/products/downloadSeriesDownNew/8083/1210>.
- [3] CUIDEVICES, "Modular incremental encoder." disponibile ONLINE: <https://www.mouser.it/datasheet/2/670/amt10-2298320.pdf>.
- [4] Amprobe, "5xp-a digital multimeter." disponibile ONLINE: <https://docs.rs-online.com/4d24/0900766b80bdd0ea.pdf>.
- [5] Turnigy, "Turnigy plush manual." disponibile ONLINE: <https://www.rcgroups.com/forums/showatt.php?attachmentid=12471557&d=1566711302>.