



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE

Sviluppo e porting di firmware per drone di tipo Ducted Fan

Sottotitolo della Tesi

Candidato:

Robert Laurentiu Mincu

Relatore:

Andrea Bonci

Correlatore:

Prof. Nome Correlatore 1

Prof. Nome Correlatore 2

Anno Accademico 2024-2025



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE

Sviluppo e porting di firmware per drone di tipo Ducted Fan

Sottotitolo della Tesi

Candidato:

Robert Laurentiu Mincu

Relatore:

Andrea Bonci

Correlatore:

Prof. Nome Correlatore 1

Prof. Nome Correlatore 2

Anno Accademico 2024-2025

UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA IN INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE
Via Brecce Bianche – 60131 Ancona (AN), Italy

La dedico a Giada, Alina e Paolo

Ringraziamenti

Grazie a Tizio e Caio

Ancona, Dal 1/2/2025 al 1/10/2025

Robert Laurentiu Mincu

Abstract

The `univpmthesis` class produces a template for the Bachelor and Master thesis manuscripts for students at UNIVPM. The class allows to write your manuscript using either italian or english; so, if you are an international student, feel free to use it!

Sommario

Indice

0.1.	STM32 NUCLEO-H745ZI-Q. La scheda di controllo	1
0.1.1.	STM32H745ZI-TQ6. Caratteristiche tecniche	1
0.1.2.	Caratteristiche della scheda NUCLEO-H745ZI-Q	4
0.2.	TATTU LiPo. Batteria ricaricabile	6
0.2.1.	TATTU LiPo. Caratteristiche fisiche.	6
0.2.2.	TATTU LiPo. Caratteristiche tecniche e considerazione operative.	6
0.3.	PowerSafe Twin ADV. MNR-electronics	7
0.3.1.	PowerSafe Twin ADV. Caratteristiche fisiche e tecniche . .	8
0.4.	Turnigy AereoDrive SK3-3536 1400KV	9
0.4.1.	Turnigy AereoDrive SK3-3536 1400 KV. Caratteristiche tecniche:	10
0.5.	Turnigy Plush 40A. <i>Electronic Speed Controller</i>	11
0.5.1.	Turnigy Plush 40A. Caratteristiche fisiche:	12
0.5.2.	Turnigy Plush 40A. Programmazione	12
0.5.3.	Turnigy Plush 40A. Risoluzione e controllo dei motori . . .	15
0.6.	VL53L1X ST. Sensore Time-of-Flight per la misurazione a lunga distanza	17
0.6.1.	Il principio di misura di un sensore <i>Time-of-flight, ToF</i> . . .	17
0.6.2.	VL53L1X STMicroelectronics. Caratteristiche fisiche	17
0.6.3.	VL53L1X STMicroelectronics. Catatteristiche tecniche . . .	18
1. Sistema		21
1.1.	Modello matematico	21
1.2.	Controllo	21
2. Hardware		22
2.1.	Componente1	22
2.2.	Componente2	22
2.3.	Schema dei collegamenti	22
3. Software		24
3.1.	Diagramma di flusso	24
3.2.	Gestione singoli componenti	24
3.2.1.	Gestione componente1	26
3.2.2.	Gestione componente2	26
3.3.	Funzionamento complessivo	26

Indice

4. Test e risultati	27
4.1. Test1	27
4.2. Test2	28
Conclusioni e sviluppi futuri	29
Appendici	29
A. Appendix1	30
B. Il mio primo capitolo con L^AT_EX	31
B.1. Introduzione	31
B.2. Organizzazione dei files	31
B.2.1. Opzioni della classe	32
B.2.2. Pacchetti di supporto	32
B.3. Brevissimi esempi sull'inserimento degli oggetti di testo	32
B.3.1. Equazioni	32
B.3.2. Tabelle	33
B.3.3. Figure	33
B.4. Inserimento della bibliografia	33
B.5. Esempio di codice in python	34
B.6. Esempio di codice in C	34

Elenco delle figure

1.	Il microcontrollore STM32H745ZI-TQ6	4
2.	La scheda di sviluppo NUCLEO-H745ZI-Q	5
3.	TATTU LiPo 4S 1300mAh 75C	8
4.	PowerSafe Twin ADV MNR-electronics. Schema dei collegamenti	9
5.	PowerSafe Twin ADV MNR-electronics	9
6.	Turnigy Aereo Drive SK3 3536 1400 KV	11
7.	Turnigy Plush 40A	12
8.	Schema dei collegamenti per la programmazione dell'E.S.C	13
9.	Turnigy Programming Card	15
10.	Field of View del VL53L1X	19
2.1.	Schema dei collegamenti	23
3.1.	Diagramma di flusso del codice sviluppato	25
4.1.	Esempio di grafico	27

Elenco delle tabelle

B.1. Esempio Tabella.	33
-------------------------------	----

0.1. STM32 NUCLEO-H745ZI-Q. La scheda di controllo

L'STM32 NUCLEO-H745ZI-Q è una scheda di sviluppo prodotta da STMicroelectronics, basata sul microcontrollore STM32H745ZI-TQ6, appartenente alla famiglia ad alte prestazioni STM32H7. Il dispositivo è stato progettato per facilitare lo sviluppo, il *debug*, e la prototipazione di applicazioni *embedded* complesse.

La NUCLEO-H745ZI-Q è stata utilizzata per sviluppo e *testing* dei *firmware* di gestione di tutte le componenti del *D.P.D.F.* Su di essa sono state caricate tutte le librerie software sviluppate con l'ausilio di STMCubeIDE.

0.1.1. STM32H745ZI-TQ6. Caratteristiche tecniche

Architettura *dual-core*

Il microcontrollore STM32H745ZI-TQ6 presenta un'architettura *dual-core* a 32-bit composta da un processore ad alte prestazioni, il **Cortex-M7**, e da un processore il cui utilizzo è consigliato per applicazioni *real-time* e *low-power tasks*. L'architettura consente l'implementazione di applicazioni *multithread*.

In questo progetto di tesi, l'esecuzione delle librerie *software* di gestione dei dispositivi interessati è affidata al **Cortex-M4**.

A seguire un elenco delle caratteristiche concernenti all'architettura *dual-core*:

- Il **Cortex-M7** opera ad una frequenza di 480 MHz.
- Il **Cortex-M4** opera ad una frequenza di 240 MHz.

Memoria

Il microcontrollore possiede una struttura di memoria complessa e stratificata.

Le memorie interne principali sono la memoria *Flash* e la *SRAM*. Inoltre, il *core M7* monta una *cache L1* e una piccola *backup SRAM* alimentabile a batteria.

Flash

La Flash interna da **2 MByte** è l'area di memoria su cui risiede il *firmware*. La memoria in questione è organizzata in *dual-bank*, che permette la programmazione/-cancellazione di una banca dati mentre si esegue sull'altra.

SRAM

Il dispositivo dispone di circa **1 MByte** di memoria dati volatile o *SRAM*, distribuita in più banchi con caratteristiche e destinazioni d'uso differenziate. In particolare:

- La *SRAM1* e la *SRAM2*, rispettivamente da 320 KByte e 384 KByte, costituiscono i blocchi principali ad alta velocità, accessibili da entrambi i *core*. Rappresentano l'area privilegiata per l'allocazione dei dati che richiedono tempo di accesso ridotti e condivisibilità tra i due processori.

Elenco delle tabelle

- Il blocco di SRAM3, da 128 KByte, è tipicamente dedicata al **Cortex-M4**, favorendo così una separazione logica delle risorse riducendo anche i conflitti di accesso.
- SRAM4, SRAM5 e SRAM6, in ordine, 128 KByte, 64 KByte e 64 KByte completano la struttura, fornendo ulteriori spazi di memoria che possono essere destinati a funzioni specifiche o a buffer ad alte prestazioni, secondo esigenze applicative.

Cache L1

Il **Cortex-M7**, a supporto dell'efficienza complessiva, integra una *cache L1* per istruzioni e dati, riducendo la letenza di accesso alla memoria non volatile e minimizzando i colli di bottiglia dovuti alla differenza tra la velocità del processore e quella della *Flash*. In tal modo le prestazioni globali del sistema risultano incrementate.

Flexible Memory Controller e Dual-mode Quad-SPI

Il *Flexible Memory Controller* è un'unità hardware che consente l'interfacciamento diretto con memorie parallele esterne al dispositivo. Il *FMC* supporta: SRAM, PSRAM, SDRAM, NOR Flash e NAND Flash. L'integrazione del *FMC* comporta un incremento nell'efficienza delle comunicazioni. Una volta configurato il *controller*, la memoria esterna entra a far parte dello spazio di indirizzi del microcontrollore mascherando la complessità del protocollo.

Il Dual Mode Quad-SPI è un'altra interfaccia dedicata alle *Flash* seriali esterne ad alta velocità. La soluzione poco fa citata permette di espandere la memoria dedicata al *firmware* o archiviare dati.

Memory Protection Unit

La *Memory Protection Unit* è una componente hardware integrata in ciascun *core* del microcontrollore. La mansione del circuito integrato in questione è quella di controllare e regolare l'accesso alla memoria.

Unità di elaborazione

L'architettura dell'STM32H745ZI-TQ6 integra anche unità di elaborazione dedicate ai calcoli numerici e al processamento dei segnali.

Floating Point Unit

La *FPU* è un'unità hardware integrata nei due *core* di elaborazione che consente l'esecuzione diretta di operazioni in virgola mobile. La presenza di questo circuito integrato riduce drasticamente la latenza di calcolo in applicazioni che richiedono elaborazioni numeriche complesse.

Nel caso del **Cortex-M7** la FPU supporta sia la *single precision* che la *double precision*, mentre il **Cortex-M4** solo una FPU a precisione singola.

Digital Signal Processing

Il *Digital Signal Processing* è un supporto hardware per istruzioni dedicate all'elaborazione numerica di segnali digitali.

Convertitori

Convertitori Analogici-Digitali

L'STM32H745ZI-TQ6 integra tre convertitori analogici digitali **SAR ADC** (*successive approximation analog-to-digital converter*) a 16-bit: ADC1, ADC2 e ADC3. I primi due, l'ADC1 e l'ADC2, sono *"tightly coupled"*, cioè strettamente accoppiati, consentendo una sincronizzazione precisa delle conversioni. Questo approccio permette di ottenere alte frequenze di campionamento, precisione temporale e riduzione del carico computazione del core. L'ADC3 invece è istanziato separatamente. La risoluzione di questi può essere configurata a 16, 14, 12, 10 o 8 bit.

PARLA DELLE VARIE MODALITÀ DI CONVERSIONE DEGLI ADC

Convertitori Digitali-Analogici

Il microcontrollore possiede 2 **DAC** indipendenti la cui risoluzione è a 12 bit.

Timer

I *timers* sono periferiche hardware fornite dal microcontrollore per svolgere attività correlate al tempo. L'STM32H745ZI-TQ6 fornisce 15 *timer* di diverso tipo e 5 *low power timers*. Alcuni di questi possono essere utilizzati per la generazione di segnali PWM.

Periferiche di comunicazione

L'STM32H745ZI-TQ6 integra un insieme ampio e variegato di protocolli di comunicazione. Le periferiche possono essere raggruppate in tre categorie principali: comunicazione seriale, comunicazione veloce e interfacce multimediali speciali. Per quanto riguarda la comunicazione seriale, il microcontrollore supporta i protocolli: *I²C*, USART/UART e SPI. Nell'ambito della comunicazione veloce e avanzata il dispositivo supporta i protocolli: CAN (*Controller Area Network*), USB OTG, Ethernet MAC, SD/SDIO/MMC [A]. L'integrazione delle interfacce multimediali consentono la gestione di applicazioni audio, grafiche e visione artificiale: SAI (*Serial Audio Interface*), DFSDM (*Digital Filter for Sigma-Delta Modulators*) e DCMI (*Digital Camera Interface*). [A].

Eventi asincroni

Il microcontrollore offre la possibilità di gestire con efficacia ogni genere di *interrupt* (evento asincrono) grazie all'integrazione di un'unità di elaborazione dedicata: il NVIC (*Nested Vector Interrupt Controller*).

Elenco delle tabelle



Figure 1.: Il microcontrollore STM32H745ZI-TQ6

0.1.2. Caratteristiche della scheda NUCLEO-H745ZI-Q

La NUCLEO-H745ZI-Q è la scheda di sviluppo (*development board*) che proietta all'esterno tutte le funzionalità del microcontrollore poco fa descritto.

ST-LINK/V3E

L'ST-LINK/V3E è un'interfaccia hardware, gestita dal microcontrollore STM32F723, dedicata al collegamento tra elaboratore esterno e microcontrollore. Il modulo permette di caricare il compilato *firmware* sul microcontrollore direttamente dalla porta USB Micro-B collegata al sistema di elaborazione. Il modulo, permette, inoltre, di eseguire il *debug* in tempo reale del codice caricato sull'*MCU*^[A].

L'ST-LINK/V3E fornisce anche interfacce ausiliarie utili ai fini dello sviluppo.

A seguire un breve elenco esplicativo di queste funzionalità aggiuntive:

- VCP (*Virtual COM Port*): lo ST-LINK crea una porta seriale virtuale via USB, in tal modo è possibile collegarsi al microcontrollore come se fosse collegato via UART classica.
 - MSC (*Mass Storage*): la scheda di sviluppo appare al sistema di elaborazione come una memoria esterna.

Alimentazione

La scheda di sviluppo offre diversi approcci per l'alimentazione:

- Alimentazione via USB Micro-B ST-LINK a 5V.
 - Alimentazione via Jack esterno da 7-12V.
 - Alimentazione tramite *pin Vin* esterno a 3.3V.

Nonostante la presenza di diversi approci di alimentazione e i differenti voltaggi, la tensione di alimentazione del microcontrollore è 3.3V Rev, ottenuta attraverso regolatori interni.

Elenco delle tabelle

Formato fisico

La scheda di sviluppo in esame appartiene alla famiglia **Nucleo-144** di STMicroelectronics e, dunque, mette a disposizione 2 connettori **Morpho** che espongono i 144 pin del microcontrollore. Il sistema di espansione poco fa citato viene chiamato ST Zio.

La NUCLEO-H745ZI-Q offre, inoltre, la compatibilità hardware e pinout con gli *Shield Arduino Uno R3*.

DESCRIVI QUALI PERIFERICHE HAI UTILIZZATO PER IL PROGETTO E PERCHé

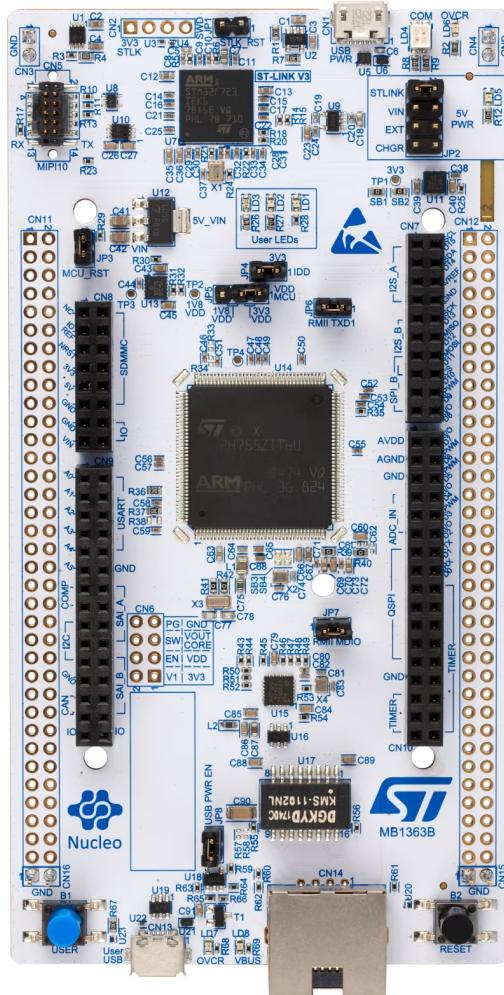


Figure 2.: La scheda di sviluppo NUCLEO-H745ZI-Q

0.2. TATTU LiPo. Batteria ricaricabile

Motori, Servomotori e E.S.C richiedono specifici livelli di tensione per il loro funzionamento, tensioni che non possono essere fornite dalla NUCLEO H745ZI-Q. Si rende, quindi, necessario l'inserimento nel progetto di batterie ricaricabili.

La selezione delle batterie disponibili sul mercato è stata condotta a seguito di un'analisi approfondita delle caratteristiche elettriche di motori, servomotori ed E.S.C, unitamente ad uno studio dettagliato delle esigenze di autonomia necessarie per le prove di volo, rapportando il tutto al *budget* disponibile.

Le **TATTU LiPo** possiedono le prestazioni richieste dal progetto.

0.2.1. TATTU LiPo. Caratteristiche fisiche.

- Dimensioni: $74 \times 35.5 \times 29$ [mm].
- Peso: 155 [g].
- Numero di celle: 4.

0.2.2. TATTU LiPo. Caratteristiche tecniche e considerazione operative.

LiPo

La sigla *LiPo*, sta ad indicare la tecnologia elettrochimica utilizzata nella costruzione della batteria. *LiPo* nasce da *Lithium-Polymer*, ovvero polimero di litio. La particolare nominazione deriva dal componente chimico utilizzato come elttrolita, che, nel caso in esame, non è liquido, ma un polimero gelificato. Malgrado nelle fonti ufficiali non venga specificato il polimero usato nel progetto delle *TATTU LiPo*, con il fine di informazione, è stato deciso di riportare quello più comunemente utilizzato per la tipologia di batterie in esame, il *polivinildenfluoro esafluoropropilene* (PVDF-HFP). Rispetto ad altri sistemi elettrochimici presenti sul mercato (come ad esempio: Li-ion, *LiFePo₄*, NiMH ecc..), le batterie LiPo presentano un'elevata densità di potenza, una bassa resistenza interna e una notevole flessibilità nelle geometrie costruttive, oltre ad un contenuto peso. Tali attributi risultano vantaggiosi nell'ambito della progettazione di UAV, come il *D.P.D.F*.

Tensione

La *TATTU LiPo* possiede 4 celle da **3.7** [V]. Le celle sono collegate in serie, perciò la tensione nominale è circa **14.8** [V].

A carica completa, ogni cella può raggiungere i **4.2** [V], emettendo un'uscita di **16.8** [V]. Il mantenimento della tensione delle celle nell'intorno dei 3.7 [V], contribuisce a preservare la stabilità chimica interna delle batterie, riducendone l'erosione delle celle e prolungandone la vita utile. Si consiglia di non mantenere le celle a 4.2 [V] per periodi prolungati di tempo, evitandone così il danneggiamento.

Nella fase iniziale e di sviluppo *firmware*, le batterie sono state tenute in bilanciamento

Elenco delle tabelle

alla tensione nominale di 14.8 [V], mentre, nella fase finale, in cui i *test* di volo sono stati protagonisti, al fine di ottenere un'aumento dell'autonomia delle batterie e della prestazione dei motori, queste venivano caricate, mantenendo il bilanciamento tra celle, fino a 4.2 [V] per cella. Infine, è necessario adottare un'ulteriore accortezza per evitare danni. È consigliabile non far scendere la tensione della batteria al di sotto dei **12 [V]**, mantendo così il valore di tensione di ogni cella sopra i **3 [V]**.

Capacità

La *capacità* di una batteria, misurata in [mAh], *milliAmpere-ora*, definisce la quantità di carica elettrica che la batteria può immagazzinare e fornire nel tempo. Nel caso in esame, la capacità è di **1300 [mAh]**, vale a dire che la batteria può fornire **1.3 [A]** per un'ora. In generale si può utilizzare la seguente formula approssimata per il calcolo del tempo di funzionamento:

$$t_h = \frac{C \text{ [mAh]}}{A \text{ [mA]}} \quad (0.1)$$

dove t_h rappresenta il tempo di funzionamento in ore, C la capacità della batteria espressa in milliAmpere-ora e A la corrente assorbita espressa in milliAmpere. Con il fine di aumentare l'autonomia di volo, si è optato per l'accoppiamento in parallelo di **due TATTU LiPo**, così da ottenere una capacità totale di **2600 [mAh]**. L'accoppiamento in sicurezza è stato effettuato utilizzando il **PowerSafe Twin della MNR-electronics**, il quale verrà approfondito nella successiva sezione.

Capacità di scarica

Con il termine *capacità di scarica* ci si riferisce alla quantità di corrente massima che la batteria può fornire, in sicurezza e in maniera continuativa, senza danneggiarsi. La *TATTU LiPo* possiede una capacità di scarica pari a **75C**, vale a dire che può erogare:

$$A_{max} = C_s \cdot C \implies A_{max} = 75 \text{ [C]} \cdot 1.3 \text{ [A]} = 97.5 \text{ [A]} \quad (0.2)$$

senza surriscaldarsi o danneggiarsi.

Nella precedente formula, A_{max} si riferisce alla corrente massima erogabile dalla batteria in sicurezza, espressa in [A], *Ampère*. C_s e C , invece, si riferiscono rispettivamente alla capacità di scarica e alla capacità di erogazione di corrente in'ora, espressa sempre in [A].

0.3. PowerSafe Twin ADV. MNR-electronics

Per garantire la sicurezza e l'affidabilità dell'alimentazione a due pacchi batteria, si è deciso per l'implementazione del **modulo di ridondanza, PowerSafe Twin ADV di MNR-electronics**.

Il dispositivo incarna un'avanzata soluzione di *ridondanza attiva*. Quest'ultima fa



Figure 3.: TATTU LiPo 4S 1300mAh 75C

riferimento alla capacità del sistema di intervenire attivamente nel funzionamento del circuito con il fine di garantire continuità di alimentazione. Il modulo, di fatti, monitora continuamente alcune variabili di stato, come tensione e corrente, selezionando, in tempo reale, il pacco alimentante il carico. Per esempio, se dovesse abbassarsi la tensione di una batteria sotto una certa soglia critica, il dispositivo commuterebbe automaticamente sull'altra, evitando così, una significativa interruzione della potenza fornita.

Nell'architettura del sistema è possibile distinguere tre moduli distinti.

Modulo di potenza

Il modulo di potenza è il circuito elettrico responsabile: della commutazione tra i pacchi batteria, del bilanciamento e della distribuzione di corrente verso il carico. Inoltre, il dispositivo implementa circuiti di protezione, isolando la sorgente guasta, qualora si dovesse presentare un malfunzionamento, ed evitando il trasferimento di corrente fra i pacchi batteria.

Modulo di controllo

Il modulo di controllo rappresenta il "centro di comando" del dispositivo. L'implementazione di un microprocessore permette la continua misura di parametri decisionali fondamentali, quali: la tensione delle batterie, la corrente di carico e l'eventuale squilibrio. L'inserimento del modulo di controllo nell'architettura circuitale, permette l'implementazione della, precedentemente citata, tecnologia di *ridondanza attiva*.

Modulo di allarme

Il modulo di allarme provvede alla segnalazione di eventuali guasti mediante emissioni luminose (LED) ed acustiche (Buzzer).

0.3.1. PowerSafe Twin ADV. Caratteristiche fisiche e tecniche

A seguire, un breve elenco delle caratteristiche fisiche e tecniche del dispositivo:

Schema di collegamento e montaggio

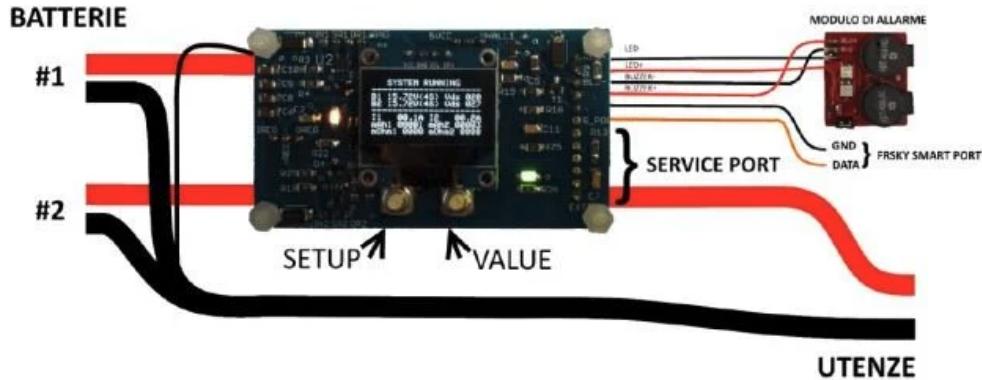


Figure 4.: *PowerSafe Twin ADV MNR-electronics. Schema dei collegamenti*

- Dimensioni con involucro protettivo: $80 \times 59 \times 26$ [mm].
- Peso, considerando sia l'involucro protettivo che i cablaggi (connettori XT60): 100 [g].
- Corrente massima gestibile: 100 [A] (in continua).
- Numero di celle per pacco batteria gestibili: $[3S - 8S]$
- Tensione massima ammessa in ingresso: ≈ 33.6 [V]. Considerando il minimo 3.7 [V] e il massimo 4.2 [V] per cella: $[11, 1 \div 33, 6]$ [V].



Figure 5.: *PowerSafe Twin ADV MNR-electronics*

0.4. Turnigy AeroDrive SK3-3536 1400KV

Il Turnigy AeroDrive SK3 3536 1400kv è un motore *Brushless* alimentato a corrente continua e pilotato a trifase con correnti alternate.

Il rotore contiene magneti permanenti disposti con polarità alternata lungo la sua circonferenza. Lo statore invece, ospita tre avvolgimenti elettrici (fasi) disposti

Elenco delle tabelle

generalmente a 120 gradi elettrici l'uno rispetto l'altro, connessi ai tre terminali d'uscita che si identificano come A,B e C.

Il rotore, essendo formato da magneti permanenti, non necessita di alimentazione: è lo statore a generare il campo magnetico rotante che induce il moto.

Il cambio di polarità degli avvolgimenti deve essere gestito da un sistema elettronico esterno, *l'Electronic Speed Controller (ESC)*, che nella successiva sezione verrà approfondito.

I tre conduttori che fuoriescono dal motore rappresentano i terminali delle tre fasi dello statore. Ciascun filo è collegato a uno degli avvolgimenti, e nessuno di essi costituisce un riferimento comune. Per far ruotare il motore è necessario che tali fasi vengano alimentate secondo una precisa sequenza temporale, generando un campo magnetico rotante. Se si dovesse scambiare due dei tre cavi, l'ordine di commutazione si invertirebbe con conseguente variazione del senso di rotazione.

0.4.1. Turnigy AeroDrive SK3-3536 1400 KV. Caratteristiche tecniche:

- Tensione operativa: $[11.1 \div 16.8]$ [V].
- Valore KV: 1400 ($\frac{RPM}{V}$).
- Potenza massima: 590 [W].
- Corrente massima: 40 [A].
- Resistenza interna: $[0.021 \div 0 - 025]$ [Ohm].
- Corrente a vuoto: 0.021 [A].
- Diametro dell'albero motore: 5 [mm].
- Peso: 111 [g].
- Spaziatura fori di montaggio: 25×25 [mm].
- Connettori: *bullet* da 3.5 [mm].
- Numero dei poli: 12.

Il valore "KV" è comune nel contesto del modellismo ed indica il numero di giri al minuto (RPM) che il motore compie per ogni *volt* applicato a vuoto, in formula:

$$RPM = KV \cdot V \quad (0.3)$$

A tensioni di alimentazione maggiori rispetto a quelle specificate il motore rischia di assorbire troppa corrente e surriscaldarsi.

I giri al minuto massimi dipenderanno dalla tensione applicata e dalle dimensioni delle eliche.

Elenco delle tabelle

Nel presente progetto, il motore in interesse deve essere alimentato a batteria, vale a dire in corrente continua, lasciando spazio alla necessità di un dispositivo che trasformi la tensione continua erogata dalle batterie in un segnale trifase a corrente alternata. Il dispositivo in questione è il precedentemente citato *Electronic Speed Controller*.



Figure 6.: Turnigy Aero Drive SK3 3536 1400 KV

0.5. Turnigy Plush 40A. *Electronic Speed Controller*

Il Turnigy Plush 40A è un controllore elettronico di velocità per motori *brushless* a corrente continua (BLDC).

Trasforma la tensione continua in ingresso in tensioni alternate sfasate di 120 gradi elettrici.

L'applicazione della corretta sequenza di correnti alle fasi dello statore richiede la conoscenza della posizione angolare in ogni istante di tempo. Il Turnigy Plush 40A stima tale informazione in maniera indiretta, misurando la forza controelettromotrice indotta dal rotore, evitando così l'uso di sensori di posizione; il dispositivo in interesse può dunque essere definito *sensorless*. L'E.S.C in questione può essere concettualmente suddiviso in due sezioni principali: la sezione di potenza e la sezione di controllo. La **sezione di potenza** è responsabile della conversione *continua-trifase*. Tale conversione è mediata da un ponte trifase a sei MOSFET, suddiviso a sua volta in tre mezzi ponti, ciascuno associato ad una fase del motore. Ogni mezzo ponte è composto da un tranzistor di alto lato e da un transistor di basso lato, collegati rispettivamente al potenziale positivo della batteria e a massa. L'opportuna commutazione di apertura e chiusura dei sei MOSFET comporta la generazione di un segnale di tensione alternata per ogni fase.

La **sezione di controllo** ha come protagonista un microcontrollore che si occupa della generazione di segnali di pilotaggio e della stima della posizione del rotore. Il microcontrollore riceve in ingresso il comando di velocità di rotazione del motore sotto forma di segnale PWM. Sulla base di tale comando, il microcontrollore determina il *duty cycle* del PWM e calcola la sequenza di commutazione dei sei MOSFET, con

Elenco delle tabelle

conseguente generazione del campo magnetico rotante dello statore.

Il *Double Propeller Ducted Fan* monta due Turnigy Plush 40A, uno per ogni motore.

0.5.1. Turnigy Plush 40A. Caratteristiche fisiche:

- Corrente nominale: 40 [A].
- Corrente di picco: 55 [A].
- Presenza di un *Battery Eliminator Circuit* (BEC): 5 [V] a 3 [A].
- Celle di batteria necessarie per il corretto funzionamento del dispositivo: [2 ÷ 6].
- Peso: 39 [g].
- Dimensioni in [mm]: $60 \times 24 \times 15$.



Figure 7.: Turnigy Plush 40A

0.5.2. Turnigy Plush 40A. Programmazione

Il Turnigy Plush 40A offre la possibilità di programmare il suo comportamento operativo, soddisfando così un vasto spettro di possibili esigenze progettuali.

Al fine di ottenere uno specifico comportamento dei motori, tale da adeguarsi alle particolari esigenze di un UAV come il *Double Propeller Ducted Fan*, entrambi gli E.S.C sono stati minuziosamente riprogrammati.

L'accesso alla *Programming Mode* e l'effettiva programmazione sono state effettuate mediante l'apposita *Turnigy Programming card*, acquistata separatamente. Per effettuare la programmazione, è necessario disporre i collegamenti elettrici come in figura:

La *Turnigy Programming Card* richiede di essere alimentata da un segnale continuo di tensione appartenente al seguente intervallo: $[4.8 \div 6][V]$. Per la navigazione fra i

Elenco delle tabelle



Figure 8.: Schema dei collegamenti per la programmazione dell'E.S.C

parametri è necessario utilizzare il pulsante *up/down*, mentre, per la selezione dei corrispondenti valori, deve essere utilizzato il pulsante *left/right*.

Nel momento in cui verrà selezionato il valore desiderato del parametro in programmazione, il LED blu, che risponde al nome di *connecting*, lampeggerà, segnalando così il successo dell'operazione. A seguire, una piccola spiegazione del significato dei parametri e i valori attualmente selezionati:

Brake

Il parametro *Brake* si riferisce alla modalità con cui viene gestita la mancanza di segnale PWM di controllo. Nel caso in cui dovesse essere impostato ad "ON" il motore verrà immediatamente frenato.

Il parametro in analisi non ha grande valenza ai fini progettuali dal momento che qualsiasi prova di volo è stata effettuata nell'apposita gabbia, per cui qualunque combinazione del parametro è accettabile. Nello svolgersi del progetto il parametro è stato impostato ad "OFF".

Battery Type

Il *Battery Type* riferisce al dispositivo la tipologia di batteria in utilizzo, questo perché L'E.S.C monitora costantemente la tensione della batteria, quando questa scende sotto una certa soglia, entra in azione il *Low Voltage Cut-off*, riducendo o interrompendo la potenza fornita al motore con il fine di evitare il danneggiamento delle celle. Diverse tipologie di batterie hanno diverse curve di scarica. L'E.S.C può gestire solamente due tipologie di batterie, LiPo e NiHM. Il progetto prevede l'utilizzo delle **Tattu LiPo** perciò il parametro è stato impostato ad Li-xx.

Low Voltage Protection Mode (Cut-off Type)

Come già riferito in precedenza, la *Low Voltage Cut Off* evita il danneggiamento delle celle delle batterie quando queste scendono sotto una certa soglia impostabile. Il dispositivo permette di scegliere tra *Soft Cut-Off* e *Cut-Off*. Se si dovesse scegliere la modalità *Soft*, al superamento della soglia, l'E.S.C ridurrà gradualmente la potenza in uscita. Se si dovesse, invece, scegliere l'altra modalità, il dispositivo interromperà immediatamente la linea di fornitura di potenza al motore.

Elenco delle tabelle

Al fine di evitare il danneggiamento del sistema, dovuto a un'interruzione improvvisa della fornitura di potenza ai motori durante il volo, seppur all'interno della gabbia, è stata impostata la modalità *Soft Cut-Off*.

Cut Off Voltage

Il parametro permette di impostare la soglia al di sotto della quale entra in azione la *Low Voltage Protection Mode*.

- *Low*: 2.6 [V].
- *Medium*: 2.85 [V].
- *High*: 3.1 [V].

Con il fine di limitare il danneggiamento delle batterie, ottenendo contemporaneamente una discreta autonomia di volo, il parametro è stato impostato a **Medium**.

Numericamente, considerando le batterie **Tattu LiPo** a 4 celle, la soglia di attivazione del L.V.C è: $2.85V \cdot 4 = 11.4V$.

Start Mode

la *Start Mode* gestisce la modalità di accelerazione del motore da fermo. Sono disponibili tre diverse modalità di partenza, pensate per adattarsi a qualsiasi applicazione. La *Normal Start Mode* costituisce la modalità di accelerazione più reattiva. L'E.S.C applica una rampa di potenza relativamente breve, nel giro di $[0.3 \div 0.5][s]$ il motore raggiunge i primi giri utili. L'appena descritta modalità è stata scartata a priori dal momento che, come conseguenza di un'accelerazione repentina, si ha forte stress meccanico sull'albero motore, inoltre l'improvvisa accelerazione delle eliche può generare forze indesiderate.

Nella *Soft Start* la rampa di ingresso è più dolce, di fatti il motore, sotto il controllo dell'E.S.C, impiega $[1 \div 1.5][s]$ per arrivare a regime. Infine, nella *Very Soft Start* la potenza viene incrementata nell'arco di $[2 \div 3][s]$.

Qualsiasi prova di volo del *D.P.D.F* è stata eseguita nell'apposita gabbia di contenimento, in cui il sistema è stato sospeso mediante funi di sostegno, perciò, con il fine di evitare movimenti indesiderati, stress meccanico ed errori di misurazione è stata impostata la *Very Soft Start*.

Timing Mode

L'E.S.C, con il fine di genesi di rotazione, genera campi magnetici rotanti commutando le tre fasi in sequenza. Le commutazioni devono avvenire in sincronia con la posizione del rotore. Il *Timing* è l'antico angolare con cui l'E.S.C commuta le fasi rispetto alla posizione stimata del rotore.

Se il *Timing* dovesse essere basso, il campo magnetico rotante sarebbe quasi allineato con il rotore, disposizione che aumenterebbe **l'efficienza ellettrico-meccanica** complessiva del sistema ma diminuirebbe coppia e velocità. Al contrario, se dovesse

Elenco delle tabelle

essere alto, si avrebbe più potenza e velocità ma anche più dispersione di calore e meno efficienza.

L'*High Timing Mode*, a cui corrisponde un anticipo angolare di $[16 \div 30]^\circ$, favorisce potenza e velocità, utile per dispositivi ad alto numero di giri o da corsa. La *Low Timing Mode*, con un'anticipo angolare di $[0 \div 7]^\circ$, garantisce una coppia stabile con minima dispersione di calore con una leggera diminuzione della velocità massima.

La *Medium Timing Mode* ($[8 \div 15]^\circ$) costituisce un buon compromesso fra efficienza e prestazioni, dunque è stata scelta ai fini progettuali.

Music Li-Po Cells

La funzionalità è puramente estetica e non influisce sulle prestazioni, per cui non è stata impostata alcuna musica.

Governor Mode

La *Governor Mode* ha lo scopo di mantenere la velocità di rotazione del motore costante indipendentemente dalle variazioni di carico o tensione della batteria. Non essendo utile ai fini progettuali, il parametro è stato impostato su "OFF".

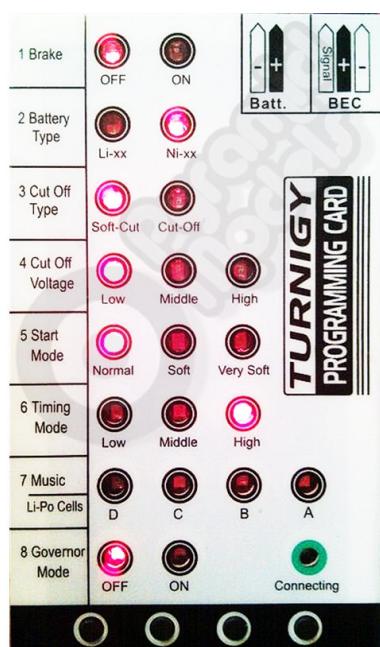


Figure 9.: Turnigy Programming Card

L'immagine riportata sopra non è riferita alla configurazione attuale degli ESC, ma ha esclusivamente scopo illustrativo.

0.5.3. Turnigy Plush 40A. Risoluzione e controllo dei motori

Il *Turnigy Plush 40A*, al fine di controllare la velocità di rotazione dei motori, richiede un segnale del tipo PWM alla frequenza tipica del modellismo, vale a dire,

Elenco delle tabelle

50[Hz].

In primo luogo, è necessaria una procedura di armamento dei motori, che consiste nell'invio del segnale PWM con *duty cycle* del 4.75%. L'esito dell'operazione di armamento può essere stabilito mediante delle segnalazioni acustiche emesse dall'E.S.C. A seguire, le emissioni acustiche più comuni:

Attesa di un segnale di armamento valido

Un'emissione ogni due secondi, segnala l'attesa di un segnale PWM idoneo all'armamento iniziale dei motori. L'emissione viene attivata dal dispositivo immediatamente dopo la connessione di questo all'alimentazione.

Esito positivo dell'operazione di armamento

Tre emissioni sonore consecutive seguite da un'emissione prolungata, indicano l'esito positivo dell'operazione di armamento.

Esito negativo dell'operazione di armamento

Una rapida ripetizione di emissioni sonore (una ogni 0.25 [s]) segnala il mancato armamento dei motori, dovuto ad un segnale di ingresso errato. Comunemente l'esito negativo di armamento si può presentare nel caso in cui il segnale PWM dovesse avere un *duty cycle* superiore al 4.75% oppure nel caso in cui il *duty cycle* dovesse repentinamente variare dal 4.75% a valori superiori. Si consiglia di attendere [2 ÷ 6] [s] prima di iniziare la variazione del *duty cycle* del segnale per il controllo della velocità dei motori.

Al seguito di vari accertamenti empirici, è stato stabilito l'intervallo di lavoro dei motori espresso in *duty cycle*: [6% ÷ 12%].

Risoluzione

Quando si tratta di E.S.C, con il termine "risoluzione" non ci si riferisce al numero di campioni di tensione logica distinguibili dal dispositivo, bensì allo "*span*" di valori del *duty cycle* distinguibili.

La conoscenza della risoluzione del dispositivo è utile nella determinazione della variazione minima del *duty cycle* del segnale PWM di controllo, che a sua volta permette lo sviluppo di un algoritmo di controllo più efficace. Per quanto riguarda il *Turnigy Plush 40A*, la documentazione ufficiale non specifica la risoluzione del dispositivo; pertanto, si è fatto riferimento a fonti non ufficiali, che indicano una risoluzione compresa tra i 8 e 12 bit.

Si è optato per uno sviluppo del controllore capace di trascurare la risoluzione del dispositivo, a causa dell'ambiguità di tale informazione.

0.6. VL53L1X ST. Sensore Time-of-Flight per la misurazione a lunga distanza

Al fine di ottenere il comportamento desiderato del **D.P.D.F.**, è necessario conoscere la distanza da terra mentre quest'ultimo è in volo. Bisogna, dunque, includere un sensore apposito.

Il **VL53L1X** di **STMicroelectronics** è pienamente coerente con i requisiti precedentemente espressi. Tuttavia, non si è optato per l'integrazione diretta del sensore della casa STMicroelectronics, si è preferito acquistare il modulo **IRIS11A0J9776** prodotto da **Pololu**. La *breakout board* di Pololu, contenente il sensore VL53L1X è stata scelta per semplificare la fase di configurazione elettronica del dispositivo.

Per comprendere il suo principio di misura, è opportuno dedicare un'intera sezione alla sua descrizione.

0.6.1. Il principio di misura di un sensore *Time-of-flight, ToF*

La modalità di misura conosciuta come *Time-of-Flight* si basa sulla determinazione del tempo impiegato da un segnale, generalmente un impulso luminoso nella banda dell'infrarosso, per compiere un viaggio di andata e ritorno tra un emettitore ed un bersaglio riflettente. In particolare, per il dispositivo in analisi, la sorgente di emissione luminosa è costituita da un laser a cavità verticale (VCSEL), che emette brevi impulsi di luce modulata. Il segnale emesso si propaga nell'ambiente fino ad incontrare un oggetto. Parte della radiazione viene riflessa e raccolta da un rilevatore sensibile alla luce, generalmente un *SPAD array* o "*Single-Photon Avalanche Diode*". Questo rilevatore è in grado di registrare fino all'arrivo di singoli fotoni, consentendo una misura estremamente sensibile del tempo di volo. Una volta noto l'intervallo temporale tra l'istante di emissione dell'impulso e quello di ricezione del suo "*eco*", il calcolo della distanza si ottiene applicando la relazione:

$$d = \frac{c \cdot \Delta t}{2} \quad (0.4)$$

dove d è la distanza dell'oggetto dalla superficie, c è la velocità della luce nel vuoto e Δt è il tempo di volo misurato. Il fattore 2 al denominatore tiene conto che il segnale percorre il tragitto due volte, andata e ritorno. I "ToF" moderni possono utilizzare tecniche avanzate di correlazione temporale o modulazione di fase per migliorare la precisione e la resistenza al rumore ambientale.

Nel caso in esame, il VL53L1X utilizza la tecnologia base ToF o dToF, "*Direct Time-of-Flight*".

0.6.2. VL53L1X STMicroelectronics. Caratteristiche fisiche

A seguire, un breve elenco delle principali caratteristiche fisiche:

- dimensioni del chip: $4.90 \times 1.25 \times 1.56$ [mm].

Elenco delle tabelle

- dimensioni del modulo: $17.50 \times 12 \times 2.56$ [mm].
- massa del chip: 30 [mg].
- massa del modulo: 0.5 [g] (senza i *pin header*).

0.6.3. VL53L1X STMicroelectronics. Catatteristiche tecniche

Alimentazione

Il sensore presenta un unico ingresso di alimentazione il cui nome è V_{DD} . L'intervallo di tensione di alimentazione è: $[2.6 \div 3.5]$ [V]

Protocollo di comunicazione

Il dispositivo è stato sviluppato per comunicare con il microcontrollore utilizzando esclusivamente il protocollo di comunicazione I^2C . Il sensore dispone della capacità di comunicare in *Fast Mode*, scambiando byte ad una frequenza di 400 [kHz].

Il sensore presenta

XSHUT e GPIO 1 (Interrupt)

Il VL53L1X dispone di un ingresso denominato **XSHUT**, che consente di spegnere o riavviare il dispositivo a livello *hardware*, indipendentemente dall'*Host*. Portando l'ingresso al livello logico basso, il sensore entra in modalità di *shutdown*, interrompendo le misurazioni e riducendo il consumo di corrente. Al contrario, applicando un livello di tensione alto, il sensore viene inizializzato e reso operativo.

Oltre che l'ingresso **XSHUT**, il sensore dispone di un'**uscita** denominata **GPIO1**, spesso indicata anche come **INT (interrupt)**, la cui funzione principale è segnalare al microcontrollore eventi o stati del sensore.

Nel presente progetto, il pin XSHUT è stato utilizzato esclusivamente per il *reset* forzato del sensore, mentre il GPIO1 è stato impiegato per segnalare la disponibilità della misura, semplificando il firmware, riducendo il carico computazionale e consentendo la gestione della temporizzazione, aspetti che verranno approfonditi nella dedicata sezione nel capitolo *Software*.

Emettitore

Il *VL53L1X* monta un emettitore **laser ad infrarossi di classe uno**. Quest'ultimo fa riferimento alla classificazione di sicurezza stabilita dalla norma internazionale **IEC 60825-1**. La classificazione ha il dovere di descrivere quanto un'emissione laser possa essere pericolosa per l'interazione umana (occhi, pelle ecc). La normativa organizza gli emettitori in *classi*, dove ogni classe rappresenta un livello crescente di rischio. Per un Laser di *classe uno*, come quello emesso dal dispositivo, le condizioni operative normali, sono considerate sicure per occhi e pelle.

Field of View (FoV)

Il *Field of View* è l'angolo massimo entro il quale il dispositivo rileva un oggetto. Per quanto riguarda l'emettitore in esame, il **FoV** è all'incirca di 27° .

A seguire, un'immagine esplicativa del *FoV*:

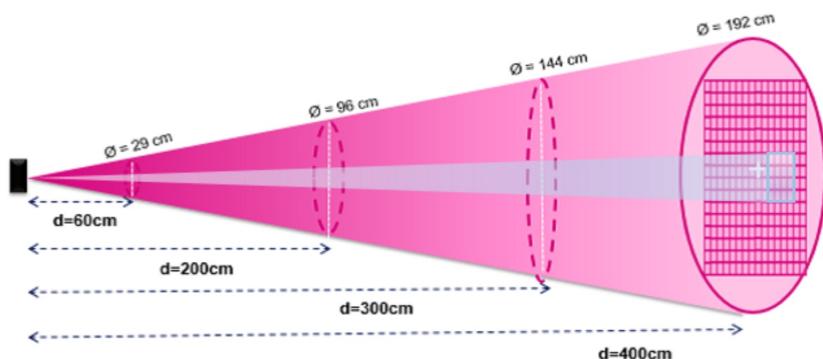


Figure 10.: Field of View del VL53L1X

Introduzione

La qualità della vita dipende sempre di più dai sistemi infrastrutturali, i quali influenzano la società contemporanea fornendo ogni tipo di bene. Tuttavia, queste sono soggette a deterioramento e, per evitare guasti o problemi più gravi, è necessario affrontare diverse problematiche nelle procedure di manutenzione. L'elevato costo, unito alla difficoltà per gli operatori umani di raggiungere i luoghi di intervento, specialmente considerando gli aspetti legati alla sicurezza, costituiscono forti limiti all'applicazione reale. Ragion per cui, i ricercatori hanno esplorato nuove modalità per automatizzare le procedure di ispezione e, attualmente [A1], la maggior parte degli **Unmanned Aerial Vehicles (UAV)** proposti per supportare le ispezioni infrastrutturali sono droni.

Nonostante le innovative soluzioni, persistono ancora importanti limitazioni da superare: queste vanno dal miglioramento delle prestazioni dei sistemi robotici, fino alle problematiche cooperative e di sicurezza. I limiti principali nelle missioni di ispezione riguardano in primo luogo le eliche dei droni, pericolose ed inadatte alle ispezioni non distruttive a contatto. Inoltre, quando questi UAV vengono impiegati in compiti cooperativi con l'essere umano, è necessaria un'interazione sicura, che non può essere garantita da droni convenzionali.

Un'ulteriore sfida consiste nel controllo preciso di questi *robot* a base flottante: in molti casi è richiesta un'elevata precisione e, attualmente, per ottenere una buona stabilità, si ricorre a soluzioni con molteplici eliche, a discapito degli elevati costi. Il **Dipartimento di ingegneria dell'informazione** dell'Università Politecnica delle Marche si è impegnato ad affrontare questi problemi presentando un'architettura

Elenco delle tabelle

alternativa di **UAV** il *Double Propeller Ducted-Fan*, considerata vantaggiosa per le ispezioni sicure e l'interazione uomo-macchina.

Il progetto, attualmente, è nella mani del laboratorio di automazione nel D.I.I dell'Università Politecnica delle Marche, a portata degli studenti dei corsi di ingegneria. L'architettura *hardware* e *software* del *D.P.D.F* non ha subito variazioni di grande portata nel tempo. Tuttavia, lo sviluppo tecnologico ha registrato notevoli avanzamenti e dunque si rende necessario un aggiornamento significativo del profilo tecnologico del drone. Ergo, il fine ultimo del progetto di tesi è stato lo "*Sviluppo e porting di firmware per un drone di tipo D.P.D.F*".

Con il termine *porting* si fa riferimento al processo di adattamento di un *firmware* sviluppato per una piattaforma *hardware* specifica, così che possa funzionare su un'altra piattaforma, diversa dalla precedente. Nel presente caso è stato eseguito il *porting* dalla scheda di controllo **Renesas Demonstration Kit YRDKRX63N** alla scheda **STM NUCLEO-H745ZI-Q**. Inoltre, sono stati apportati degli aggiornamenti al comparto servomotori e sensori. I precedenti '*servos*', **DS3115 MG Digital Servo**, sono stati sostituiti con gli **Hitec HS-82MG**, di minor peso. In merito ai sensori, il pregresso modulo di misura inerziale (I.M.U) **MPU6050 InvenSense** è stato aggiornato con il **BNO-055 Bosch Sensortec**, integrante di un microcontrollore deputato all'esecuzione di un algoritmo di fusione sensoriale [A2].

Capitolo 1.

Sistema

NOTA valido per ogni capitolo: inserire all'inizio di ogni capitolo una brevissima descrizione del contenuto del capitolo stesso.

NOTA: capitolo opzionale. Solo se necessario per comprendere quanto svolto, inserite un capitolo dedicato al sistema su cui avete lavorato (es drone quadrirotore, ballbot ..) dove descrivete ad esempio come è fatto il sistema, come si modella matematicamente e come si può controllare. Nel capitolo inserite solo le informazioni che avete effettivamente utilizzato per svolgere il task assegnatovi (**evitate il copia e incolla dalle vecchie relazioni**). Mettete la fonte da cui avete preso le informazioni (es vecchia relazione [?]).

1.1. Modello matematico

1.2. Controllo

Capitolo 2.

Hardware

Inserire sempre un capitolo "Hardware" dove si descrivono tutti i componenti hardware utilizzati (schede, sensori, attuatori..). Dedicare un paragrafo ad ogni componente. Per ogni componente riportare le seguenti informazioni: modello specifico, riferimento bibliografico con link ad un sito web con la documentazione del componente (es [?]), immagine, pinout, solo le informazioni rilevanti per lo svolgimento del task.

2.1. Componente1

2.2. Componente2

2.3. Schema dei collegamenti

Inserire uno schema dei collegamenti analogo a quello in Figura 2.1. Si suggerisce per lo scopo di usare il software "draw.io". Nello schema riportare tutti i collegamenti tra i componenti, specificando quale pin di un componente1 si collega con quale pin di un componente2 (per la maggior parte dei collegamenti dovete cioè avere una linea con 2 label).

Inserire anche una descrizione testuale dello schema.

Capitolo 2. Hardware

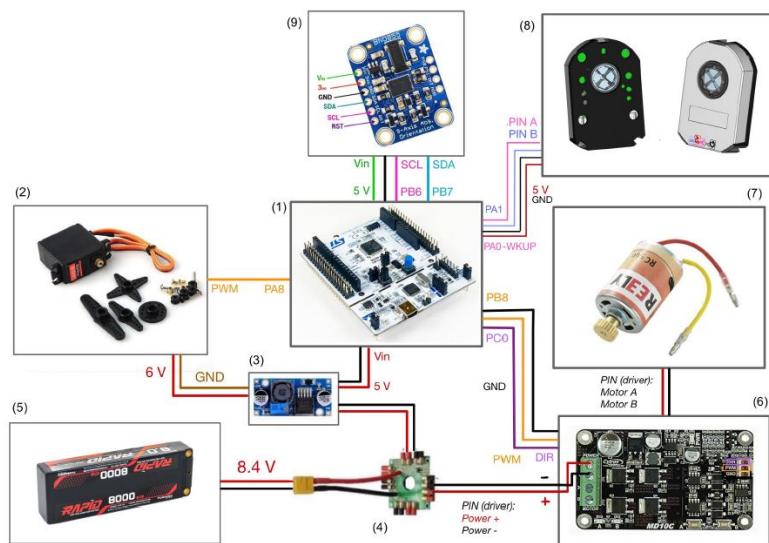


Figure 2.1.: Schema dei collegamenti

Capitolo 3.

Software

Nell'introduzione al capitolo specificate anche la versione dell'STM32CubeIDE e dell'STM32CubeMX che avete usato (se avete cambiato versione nel corso del progetto mettete l'ultima, quella su cui è sviluppato il codice che consegnate).

3.1. Diagramma di flusso

Inserire un diagramma di flusso analogo a quello in Figura 3.1 che spiega il funzionamento generale del codice sviluppato. Inserire anche una descrizione testuale del funzionamento.

3.2. Gestione singoli componenti

Dedicare un paragrafo ad ogni componente in cui viene spiegato il codice che lo gestisce. Inserite prima una descrizione generale di come si gestisce un componente di quel tipo (aiutandovi con schemi o altro). Poi le configurazioni che sono state fatte sul .ioc per poterlo utilizzare (quale periferiche sono state abilitate, come sono state configurate.. mettete degli screen del .ioc). Infine inserite e spiegate le parti di codice che lo gestiscono.

NOTA per il codice: per rendere il codice modulare e riutilizzabile è buona prassi non caricare troppo il main ma creare una libreria per ogni componente. Quindi il componente1 avrà un header file "componente1.h" e un source file "componente1.c", nel source file sono implementate le funzioni che gestiscono il componente, nell'header file ci sono i prototipi di tali funzioni in modo che esse possano essere usate nel main includendolo.

NOTA per il codice: non inserire mai nel codice dei parametri numerici senza contesto ma renderli delle costanti definite (usando la direttiva %define). Se sono delle costanti relative ad uno specifico componente vanno inserite nel relativo header file.

NOTA per il codice: per inserire il codice, anziché utilizzare screenshot utilizzate i seguenti comandi.

Per codice scritto in latex (con linguaggio Python):

Listing 3.1: "Codice in Python"

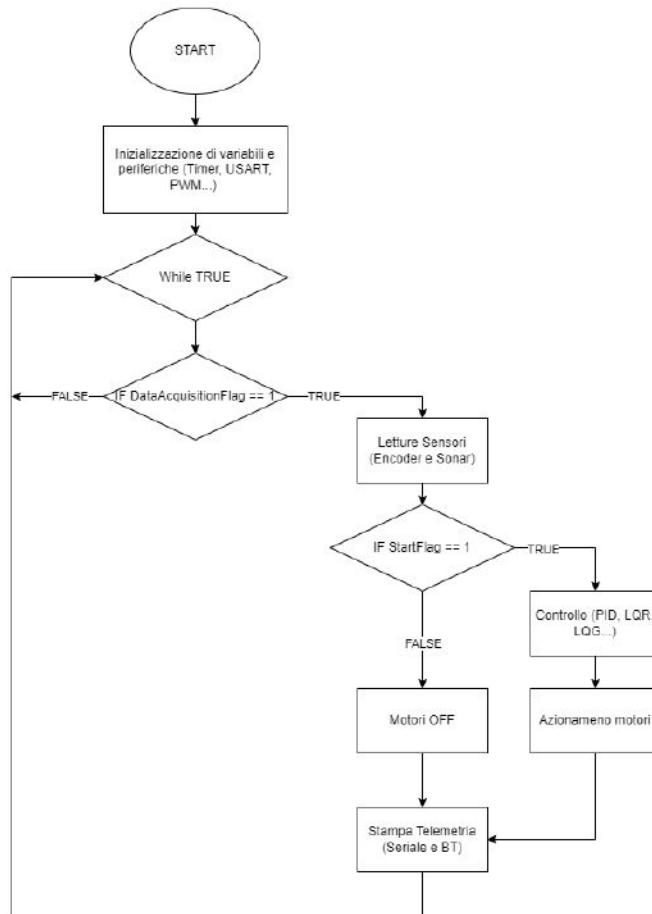


Figure 3.1.: Diagramma di flusso del codice sviluppato

```

import numpy as np

def incmatrix(genl1 ,genl2):
    m = len(genl1) # The length of the first array
    n = len(genl2) # The length of the second array
    sum = 0

    # Compute
    for i in range(0 ,n):
        for j in range(0 , m):
            sum += genl1 [n]* genl2 [m]

    # Print
    print( "The sum is " ,sum)

    return M

```

Funziona anche per codice MATLAB:

```
1 %% PREPARE WORKSPACE
2 close all
3 clearvars
4 clc
5
6 %% OPERATIONS
7 sayhello;
8
9 %% FUNCTIONS
10 function sayhello
11     fprintf("Hello world!");
12 end
```

Listing 3.2: "Codice in MATLAB"

3.2.1. Gestione componente1

3.2.2. Gestione componente2

3.3. Funzionamento complessivo

Dopo aver spiegato il codice che gestisce i singoli componenti, inserire e commentare le porzioni di codice relative al funzionamento complessivo del programma.

Capitolo 4.

Test e risultati

Dedicate un capitolo a tutti i test effettuati con relativi risultati, includete sia i test finali relativi al funzionamento complessivo, sia i test delle singole parti (se significativi).

Descrivete in dettaglio le condizioni in cui sono stati svolti i test in modo che siano ripetibili. Durante i test fate dei video e acquisite i dati (es usando la funzione di log di Putty), per i test più rilevanti è opportuno consegnare anche questo materiale per documentare i test effettuati. Nella relazione riportate i risultati con dei grafici come quello in Figura 4.1 (inserite sempre nei grafici le label sugli assi con grandezza e relativa unità di misura). Commentate in modo critico i risultati ottenuti, evidenziando sia quelli positivi sia quelli negativi.

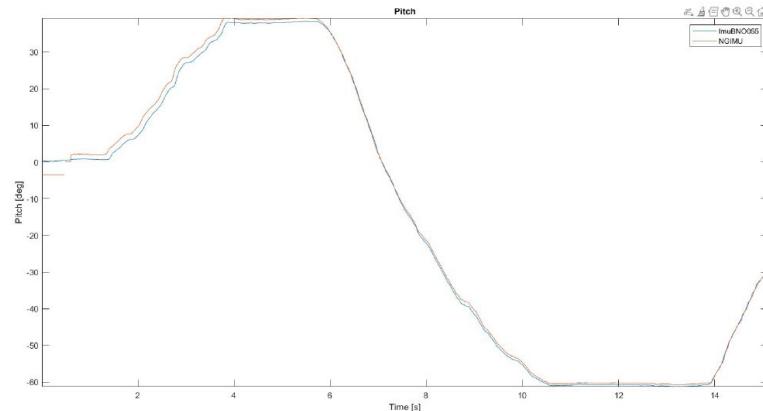


Figure 4.1.: Esempio di grafico

4.1. Test1

Per ogni test riportate: cosa state testando, in che condizioni si è svolto il test, riferimenti a video/file di log che consegnate insieme a relazione e codice, grafici dei risultati, commento dei risultati.

4.2. Test2

Conclusioni e sviluppi futuri

Riassumere brevemente il lavoro svolto rispetto al task assegnato, evidenziando quali risultati sono stati raggiunti e quali no. Se ci sono aspetti del task non completati spiegare quali sono stati i problemi riscontrati in merito.
Inserire considerazioni personali su possibili sviluppi futuri dell'attività svolta (idee per migliorarla che non avete avuto modo di sperimentare, aspetti che suggerite di approfondire, problemi da risolvere..).

Appendice A.

Appendice1

Se necessario ricorrete alle appendici per spiegare le parti "di contorno" dell'attività svolta e/o ciò che non riuscite ad inserire nello schema generale dei capitoli della relazione (es acquisizione dei dati con Matlab).

Appendice B.

Il mio primo capitolo con L^AT_EX

B.1. Introduzione

Benvenuto! Se sei alle prime armi con il linguaggio L^AT_EX, di seguito troverai le informazioni di base per iniziare la scrittura della tua tesi e per capire come organizzarla. Per un maggiore approfondimento, si consiglia la lettura di due guide (in italiano): *L'arte di scrivere con L^AT_EX* e *L^AT_EX per l'Impaziente* di Lorenzo Pantieri. Inoltre, sul web sono presenti un gran numero di forum dedicati: con una semplice ricerca è possibile esaurire ogni curiosità e risolvere qualsiasi problema.

B.2. Organizzazione dei files

Come puoi notare da questa demo, nella cartella di lavoro sono presenti due file: il primo, chiamato `univpmthesis.cls`, definisce tutti i comandi della classe, e gestisce tutte le feature grafiche e di formattazione che caratterizzeranno la tua tesi. Non modificarlo se non sei un utente esperto!

Il file `UNIVPMthesis.tex` invece rappresenta il file principale (in gergo chiamato **main file**) da cui iniziare la scrittura del tuo testo. Fondamentalmente, questo gestisce l'organizzazione del tuo lavoro di tesi e contiene tutte le informazioni per la corretta compilazione del manoscritto. È qui, infatti, che andrai a definire la classe del documento, il suo preambolo, e la successione dei capitoli e del materiale iniziale e finale.

Noterai che nel main file non è esplicitato il contenuto di ogni singolo capitolo, bensì vengono importati i dati di scrittura da files .tex secondari con il comando `input`. Tale approccio risulta particolarmente comodo di fronte alla gestione di elaborati molto lunghi, in cui ciascun capitolo può essere scritto in un file .tex dedicato.

Tutti i capitoli che costituiscono il corpo della dissertatione si trovano nella cartella `chapters`, per ognuno dei quali è stata creata una sottocartella ad hoc. Ciascuna delle sottocartelle conterrà dunque il corrispondente documento del capitolo e le immagini che vengono incluse nello stesso. Questo modo di procedere, sebbene possa sembrare laborioso, è di enorme aiuto qualora ti trovassi a gestire un gran numero di immagini e grafici. L'unica accortezza consta nel dichiarare la locazione dei file immagine

che si vogliono includere nel capitolo all'inizio dello stesso mediante il comando `graphicpath` (un esempio è dato alla primissima riga del file `chapter1.tex`).

Analogamente, la cartella `frontbackmatters` racchiude tutto ciò che non costituisce il corpo del testo, come la Bibliografia ed il capitolo di Appendice.

Poichè il template deriva dalla classe KOMA-script `scrbook`, la dichiarazione del capitolo segue sia il tradizionale comando `chapter` sia il comando `addchap`. Quest'ultimo, infatti, è utile qualora si volesse inserire un capitolo (come l'Introduzione o le Conclusioni) senza la numerazione progressiva automatica.

B.2.1. Opzioni della classe

Poichè `univpmthesis.cls` si basa su `univpmphdthesis.cls`, le opzioni offerte dalla classe sono le seguenti:

a4print definisce le dimensioni del foglio di scrittura in formato A4, permettendo all'utente di controllare meglio l'occupazione degli spazi. Il margine sinistro varia da pagina pari a dispari per tener conto dello spazio extra necessario per la rilegatura;

italian, english definisce la lingua di stampa del documento;

lof include la lista delle figure;

lot include la lista delle tabelle;

oneside, twoside definisce l'impaginazione su singola pagina (`oneside`) o su fronte-retro (`twoside`).

Nota bene: non vi è alcuna disposizione od obbligo su tale impostazione (almeno ad oggi!); è, dunque, fortemente consigliata la selezione del formato `twoside`. *D'altronde, ciò che ha valore del tuo manoscritto sono i contenuti, non il volume della tesi una volta stampata!*

B.2.2. Pacchetti di supporto

Nel preambolo sono stati inclusi i pacchetti più utilizzati per la scrittura di un documento di tesi, puoi aggiungere ulteriori pacchetti in base alle tue esigenze.

B.3. Brevissimi esempi sull'inserimento degli oggetti di testo

B.3.1. Equazioni

Come puoi osservare, Eq. B.1 rappresenta un semplice esempio di come scrivere un'equazione numerata:

$$\sigma = \sigma(\varepsilon, \xi_i) \tag{B.1}$$

B.3.2. Tabelle

Le tabelle costituiscono una categoria di oggetti flottanti, ovvero oggetti di testo che possono essere "spostati" dal compilatore al fine di garantire la massima leggibilità e chiarezza dell'intero testo. Tabella B.1 ne è un rapido esempio, e si raccomanda di posizionare la didascalia della tabella in alto rispetto alla stessa.

Table B.1.: Esempio Tabella.

Test	Campioni	Risultati
a	1	2
b	4	5

B.3.3. Figure

Le figure sono il secondo tipo di oggetto flottante che viene frequentemente utilizzato nella scrittura della tesi; di seguito è ripostato un esempio di figura singola (Figura ??), e di sottofigure (Figura ?? e Figura ??).

Nel caso di figure e sottofigure, la didascalia è posta sotto l'immagine a cui si riferisce.

B.4. Inserimento della bibliografia

L'inserimento della bibliografia rappresenta, con buona probabilità, una delle operazioni più criptiche per un utente alle primissime armi con L^AT_EX, soprattutto se si è prossimi alla cosegna del lavoro di tesi. A tal proposito, il main file qui riportato è già organizzato per accogliere qualsiasi file di bibliografia, l'unica operazione richiesta è la creazione della lista dei riferimenti mediante un file in formato .bib. Per rendere tale compito più semplice possibile, il consiglio è quello di impiegare tool e software dedicati come *JabRef* - gratuito - o *Mendeley* (esistono però molti altri References Manager software!). In questo modo dovrai preoccuparti solamente di inserire correttamente campi relativi alle pubblicazioni che vuoi inserire, il software produrrà in uscita il file già formattato per essere letto dal main file.

A questo punto puoi richiamare nel testo la citazione mediante il comando `cite` facendo riferimento alla bibtexkey che identifica il documento desiderato. In questo modo il file .bib si comporta come un database di riferimenti bibliografici: nel testo verranno inclusi solo quelli effettivamente richiamati con il comando `cite`.

Un esempio: come riportato da Von Mises in [?] ... tale teoria è confermata anche dagli studi in [1, ?].

Un ultimo dettaglio: può capitare che bibliografia e references non vengano compilate immediatamente dopo esser state inserite. Per risolvere questo problema basta compilare il main file due volte consecutive: la prima per far "leggere" al

compilatore le modifiche effettuate e stabilire quali sono i documenti invocati nel testo, la seconda per produrre la modifica sul file .pdf di output.

B.5. Esempio di codice in python

Listing B.1: "Codice in Python"

```
import numpy as np

def incmatrix(genl1, genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m, 1), int) #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2), 1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r, c] = np.where(M2 == M1[i, j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

    if M is None:
        M = np.copy(VT)
    else:
        M = np.concatenate((M, VT), 1)

    VT = np.zeros((n*m, 1), int)

    return M
```

B.6. Esempio di codice in C

```
1
2 void main(int n1){
3     int a = 1;
```

Appendice B. Il mio primo capitolo con L^AT_EX

```
4     int b = 2;  
5     return a+b*n1;  
6 }
```

Listing B.2: "Codice in C"

Buona fortuna per il tuo lavoro!

Bibliografia

- [1] R. Hill. A theory of yielding and plastic flow of anisotropic metals. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 193(1033):281–297, 1948.