



Università degli Studi di Padova

Dipartimento di tecnica e gestione dei sistemi industriali

Corso di Laurea triennale in ingegneria
meccatronica

Controllo del volo di mini quadricotteri con
scheda STEVAL-FCU001V1: studio
dell'architettura e analisi del firmware

Relatore: Prof. Antonello Riccardo

Laureando: Ongaro Daniel

Matricola: 1190133

ANNO ACCADEMICO 2022/2023

INDICE

1. INTRODUZIONE	3
2. QUADRIROTORI	4
2.1 FUNZIONAMENTO DEL VOLO	4
3. STRUTTURA E COMPONENTISTICA	6
3.1 MOTORI	6
3.2 FRAME	7
3.3 BATTERIA	7
3.4 SCHEDA STEVAL-FCU001V1.....	7
3.4.1 MICROCONTROLLORE	9
3.4.2 SENSORE DI PRESSIONE.....	11
3.4.3 MAGNETOMETRO 3D	11
3.4.4 MODULO BLUETOOTH.....	12
3.4.5 COLLEGAMENTI SULLA SCHEDA	12
4. FIRMWARE	16
4.1 RICEVITORE R/C E DECODER PPM	17
4.1.1 PROCEDURA DI CALIBRAZIONE DEI SENSORI	18
4.1.2 PROCEDURA DI ABILITAZIONE DEI MOTORI.....	19
4.1.3 CALIBRAZIONE E ABILITAZIONE NEL SISTEMA BLUENRG	20
4.2 TARGET ATTITUDE	21
4.3 MEMS SENSOR	22
4.4 ATTITUDE & HEADING REFERENCE SYSTEM (AHRS)	23
4.5 CONTROLLORE DI VOLO CON PID	24
4.5.1 IMPLEMENTAZIONE DEL PID NEL FIRMWARE.....	25
4.6 FILE main.c.....	26
5. CONCLUSIONI	28
BIBLIOGRAFIA.....	29

1. INTRODUZIONE

Lo scopo della tesi è di illustrare e spiegare la struttura e il firmware del mini quadricottero progettato e prodotto dall'azienda STMicroelectronics. STMicroelectronics è uno dei più grandi produttori mondiali di componenti elettronici, usati soprattutto nell'elettronica di consumo, nell'automotive, nelle periferiche per computer, nella telefonia cellulare e nel settore industriale.

Il design del mini quadricottero proposto da STMicroelectronics consiste in un telaio (frame) a "X" su cui sono montati i 4 motori (a corrente continua) per la propulsione, la scheda di controllo del volo (STEVAL-FCU001V1) e la batteria.

La tesi è organizzata come segue: Nella prima parte ci si concentra sulla struttura e i componenti della scheda di controllo del volo (Flight Controller Unit - FCU), inclusi i vari sensori (accelerometro, giroscopio, magnetometro e barometro) e l'unito microcontrollore (STM32F401).

Nella seconda parte ci si concentrerà invece sul firmware STSW-FCU001 appositamente progettato per la scheda elettronica STEVAL-FCU001V1, il quale consente di controllare il volo di quadricotteri radiocomandati di piccole/medie dimensioni.

2. QUADRIOTORI

I quadricotteri sono degli aeromobili costituiti da quattro rotori, generalmente progettati per volare senza pilota a bordo e per questo vengono chiamati APR (aeromobile a pilotaggio remoto) UAV (Unmanned Aerial Vehicles), o più semplicemente droni. I quadricotteri utilizzano dei sensori per monitorare la propria posizione e orientamento ed un algoritmo di controllo per la stabilità del volo. Data la relativa facilità nella manovrabilità, sono largamente utilizzati sia in ambito militare sia in ambito civile [1].

2.1 FUNZIONAMENTO DEL VOLO

Un drone è un veicolo aereo dotato di 6 gradi di libertà.

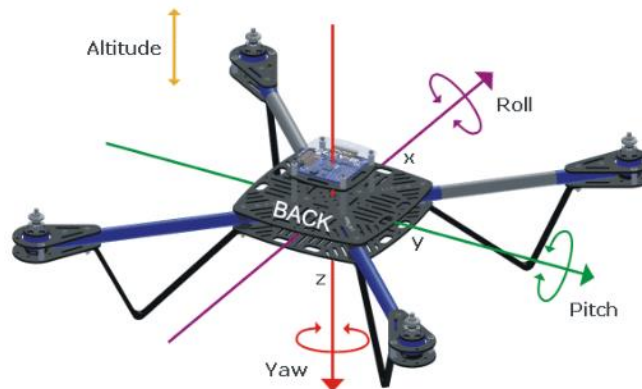


Figura 1 Rappresentazione delle assi di movimento possibili

I 6 gradi di libertà sono (vedi anche Figura 1):

1. X: movimento frontale
2. Y: movimento laterale
3. Z: movimento verticale
4. Beccheggio (pitch): oscillazione attorno ad un asse trasversale
5. Rollio (roll): oscillazione attorno ad un asse longitudinale
6. Imbardata (yaw): oscillazione attorno ad un asse verticale

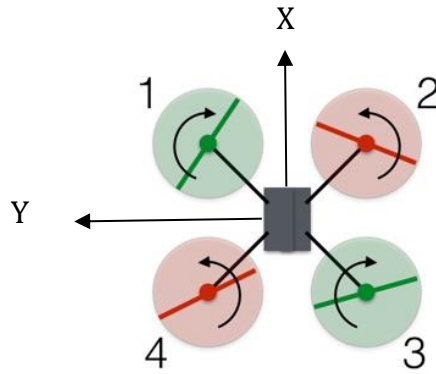


Figura 2 Rappresentazione del senso di rotazione dei motori

Nei droni i motori posti lungo lo stesso asse hanno lo stesso senso di rotazione mentre l'asse opposto presenta una rotazione contraria. Ciò produce sui due assi una coppia di momenti che, a pari velocità dei motori, si eguaglia impedendo che il drone cominci a ribaltarsi. Un incremento o decremento della velocità di un singolo motore, o di una coppia di questi, provoca uno sbilanciamento di forze o momenti tali da provocare il movimento del drone. In particolare, si ha che:

- Un incremento (decremento) della potenza in tutti e quattro i motori, provoca una salita (discesa) verticale in quota del drone.
- Un incremento di potenza di due motori dallo stesso lato genera una rotazione attorno all'asse trasversale o longitudinale. Questa manovra permette di eseguire il pitch e il roll, mantenendo inalterato lo yaw. Un aumento della velocità dei motori M2-M3 oppure M1-M4 provoca una rotazione lungo l'asse longitudinale, ovvero un roll. Per poter eseguire il pitch, ovvero una rotazione lungo l'asse trasversale, bisognerà aumentare la velocità della coppia dei motori M1-M2 o M3-M4.
- Un incremento di pari potenza di due motori posti sullo stesso braccio, porta il drone a ruotare rispetto all'asse verticale di yaw, per ottenere questo movimento quindi si dovrà aumentare la velocità dei motori M2-M4 o M1-M3. La rotazione sarà concorde al senso di rotazione dei due motori che in quel momento erogano più potenza.

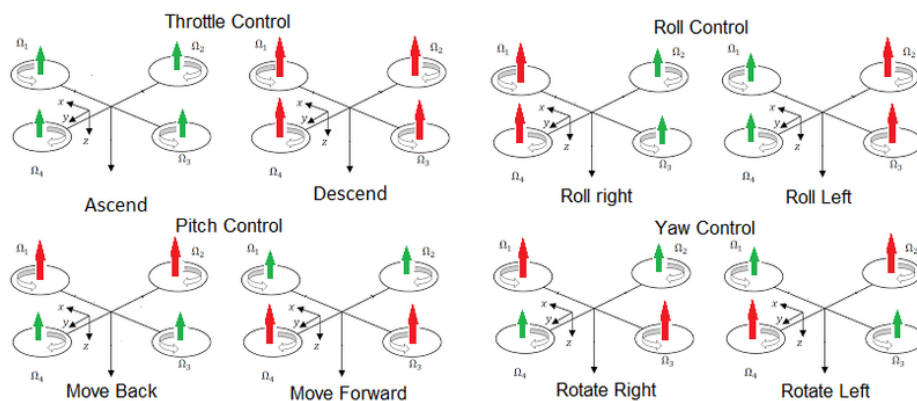


Figura 3 Funzionamento dei motori per generare un movimento di ascesa/discesa, roll, pitch e yaw

3. STRUTTURA E COMPONENTISTICA

Il mini quadricottero descritto in questo capitolo è di facile realizzazione, in quanto costituito da un numero limitato di componenti di facile reperibilità in commercio. Inoltre, il frame è realizzato mediante tecniche convenzionali di stampa 3D.

I componenti che caratterizzano la struttura del mini quadricottero sono:

- Motore DC
- Frame
- Batteria
- Scheda di controllo del volo (Flight Control Unit - FCU) STEVAL-FCU001V1

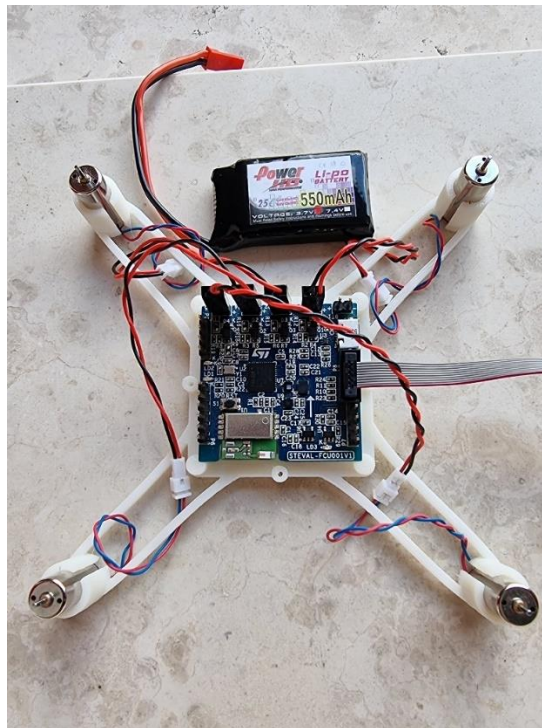


Figura 4 Mini quadricottero

3.1 MOTORI

I motori utilizzati per il mini quadricottero sono dei semplici motori in corrente continua (c.c.) con una alimentazione a 3.7 V. Una soluzione basata su motori in c.c. è più adatta per droni di piccola taglia, in quanto l'elettronica di controllo è relativamente semplice (specialmente se paragonata a quella richiesta per droni con motori brushless AC), e quindi può essere facilmente integrata sulla board di controllo del volo (con conseguente risparmio di spazio e peso).

Ad ogni modo, è possibile passare ad una configurazione più tradizionale con motori brushless AC, purché il drone venga equipaggiato con dei controlli di velocità (Electronic Speed Controller - ESC) per i motori brushless.

In tale situazione, la scheda di controllo del volo genera dei riferimenti di velocità per gli ESC, e questi ultimi alimentano i motori, regolando la velocità al riferimento imposto.

3.2 FRAME

Il frame che supporta tutti i componenti del mini quadricottero è stato progettato dalla azienda STMicroelectronics che ha fornito il codice necessario per poterlo stampare in 3D. ciò permette all'utente una grande libertà di modifica di tutta la struttura, essendo che quello fornito è solo una struttura molto base che permette l'installazione dei componenti strettamente necessari al suo funzionamento base.

Il frame è composto da una parte centrale in cui è possibile incastrare la scheda STEVAL-FCU001V1, e da quattro bracci che terminano con un incastro per i motori. Il design di tutta la struttura proprio per la sua semplicità è caratterizzato da una fragilità non indifferente, che in futuro potrebbe essere modificata implementando dei supporti per i bracci che a causa delle varie sollecitazioni potrebbero arrivare a rompersi.

3.3 BATTERIA

La batteria consigliata in questo tipo di applicazione è di tipo LiPo, che sono delle batterie ai polimeri di litio, con un valore abbastanza alto per quanto riguarda il rateo di scarica della corrente (in genere viene indicato con la lettera C). Con un valore alto di C si avrà la possibilità di richiedere rapidamente un ammontare di corrente relativamente elevato, così che se, dai motori viene richiesta una certa corrente la batteria gliela può fornire.

Per queste ragioni si è deciso di utilizzare una batteria con un valore di 25C e con una capacità di 550 mAh con una tensione di 3.7 V.

3.4 SCHEDA STEVAL-FCU001V1

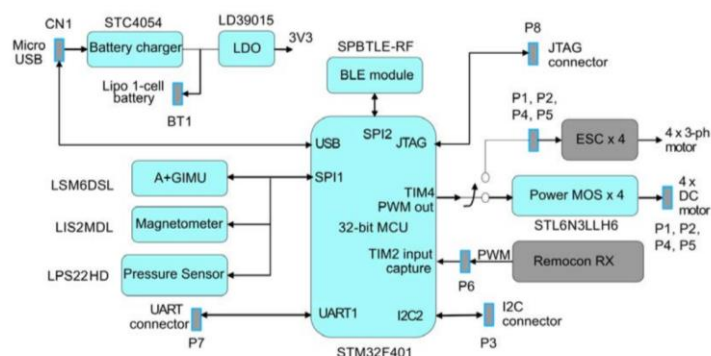


Figura 5 Schema funzionale a blocchi della scheda STEVAL-FCU001V1

La scheda elettronica STEVAL-FCU001V1, progettata da STMicroelectronics, è il cuore e cervello del mini quadricottero. Perché su di essa sono presenti tutti i

componenti necessari per il controllo e il funzionamento dei dispositivi a cui è collegata.

La programmazione della scheda elettronica è stata effettuata utilizzando i software STM32CubeMX e Keil uVision5.

- STM32CubeMX è il software che permette la programmazione dei piedini e delle varie periferiche del microcontrollore montato a bordo della scheda.
- Keil uVision5 è il software che permette la vera e propria programmazione della scheda. In quanto al suo interno tramite il linguaggio di programmazione C è possibile eseguire diverse funzioni e operazioni di controllo.

I principali componenti montati sulla scheda sono:

1. Microcontrollore: STM32F401
2. Accelerometro e giroscopio 3D: LIS2MDL
3. Sensore di pressione: LPS22HD - (260 -1260 hPa)
4. Magnetometro 3D: LIS2MDL - (± 50 gauss magnetic dynamic range)
5. Modulo Bluetooth: SPBTLE-RF v4.1

Essi consentono di ricevere le informazioni necessarie e di elaborarle per quanto riguarda il controllo di un quadricottero.

Il sensore di pressione e il magnetometro nel firmware non sono utilizzati, però sono sempre disponibili per future implementazioni da parte dell'utente [2].

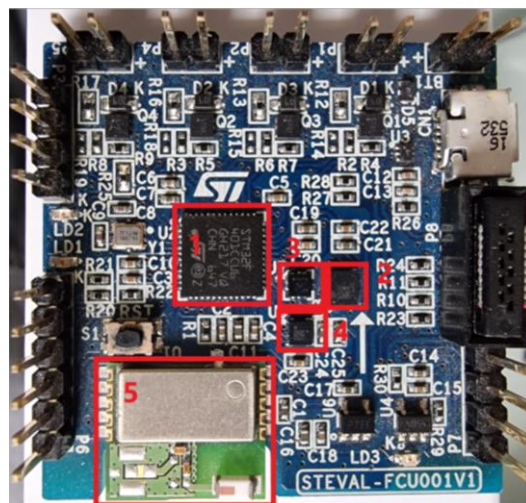


Figura 6 Scheda STEVAL-FCU001V1 con i suoi componenti principali messi in risalto

3.4.1 MICROCONTROLLORE

Il microcontrollore, detto anche MicroController Unit (MCU), è un componente che utilizza tecniche di microelettronica per contenere in un unico piccolo chip (package) diversi componenti. Tra questi troviamo:

- La CPU (ovvero un microprocessore);
- Una memoria dati volatile (la RAM);
- Una memoria di programma non volatile (Flash);
- Uno o più timer;
- Un oscillatore;
- Porte di ingresso - uscita (I/O)

Esistono anche estensioni applicabili, da implementare a seconda delle esigenze, come interfacce analogiche, porte di comunicazione o interfacce di visualizzazione e controllo (ad esempio LCD). Affinché il microprocessore sia perfettamente operativo è necessario programmarlo con un insieme di istruzioni specifiche che gli indichino le attività da svolgere [3].

Nello specifico il microcontrollore utilizzato è STM32F401, questo microcontrollore è basato sul Arm® Cortex® -M4 32-bit RISC core che opera a una frequenza massima di 84MHz. Il core Cortex®-M4 al suo interno è presente l'unità di elaborazione dei floating point (FCU), esso supporta tutte le istruzioni di elaborazione dati e i tipi di dati a precisione singola.

Il microcontrollore offre in oltre un ADC a 12 bit, un RTC a basso consumo, sei timer generici a 16 bit, tra cui un timer PWM per il controllo dei motori, due timer generici a 32 bit. Sono inoltre dotati di interfacce di comunicazione standard e avanzate.

Queste caratteristiche rendono il microcontrollore STM32F401 adatto a un'ampia gamma di applicazioni [4].

3.4.2 Inertial Measurement Unit (IMU)

L'LSM6DSL è una unità di misura inerziale (Inertial Measurement Unit - IMU) che integra al suo interno un accelerometro digitale a tre assi con un range programmabile ($\pm 2/\pm 4/\pm 8/\pm 16g$), e un giroscopio digitale a tre assi a basso consumo (0.65mA in modalità alte prestazioni) e range programmabile ($\pm 125/\pm 245/\pm 500/\pm 1000/\pm 2000$ dps) [7].

Il principio di funzionamento di questi sensori viene brevemente illustrato di seguito:

Accelerometro:

L'accelerometro è un dispositivo in grado di rilevare e misurare un'accelerazione. Il suo principio di funzionamento al livello base è molto semplice, si basti immaginarlo come un cubo con al centro una sfera sospesa da 3 molle che la attraversano, agganciate a loro volta al centro di ogni faccia del cubo.

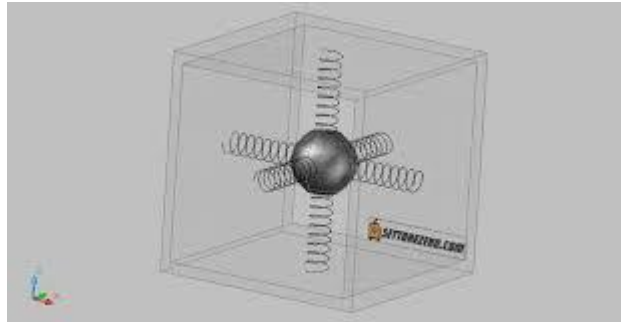


Figura 7 Schema ideale di un accelerometro a 3 assi

Muovendo il cubo nello spazio con una accelerazione diversa da 0, la sfera si muove al suo interno allungando e comprimendo le molle che la tengono sospesa. La misurazione del grado di compressione delle molle permette di stabilire l'entità dell'accelerazione nella direzione delle molle. Nella realtà, il comportamento dell'accelerometro può essere rappresentato dalla Figura 8.

Si può vedere che a causa della forza di gravità, la sfera comprime la parte inferiore della molla orientata lungo l'asse Z. Quindi, l'asse Z, subisce un'accelerazione costante verso il basso pari all'accelerazione gravitazionale terrestre $G = 9.81\text{m/s}^2$

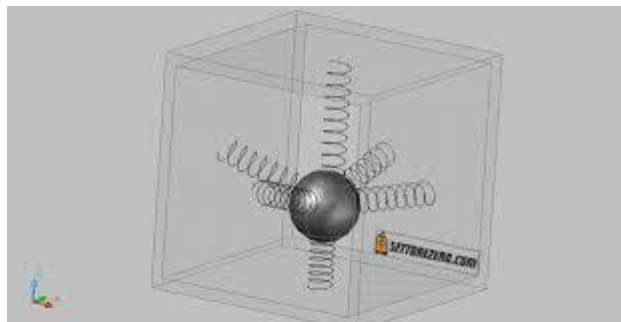


Figura 8 Schema reale di un accelerometro a 3 assi

Giroscopio:

Il giroscopio è un sensore utilizzato per la misura delle velocità angolari attorno ai tre assi principali del sensore.

I giroscopi meccanici tradizionali sono costituiti da un rotore che ruota intorno al proprio asse. Quando il supporto ruota, l'asse del rotore mantiene la sua direzione, opponendosi ad ogni azione che tende a produrre un cambiamento.

I giroscopi micro-meccanici MEMS funzionano invece secondo un altro principio, che è quello dei giroscopi vibrazionali.

In tali giroscopi è presente una massa che oscilla per esempio a 10kHz. La massa è sospesa in un sistema a molla, la lettura avviene tramite un sistema capacitivo come negli accelerometri. Quando il giroscopio viene ruotato, la rotazione esercita una forza di Coriolis perpendicolare sulla massa, che risulta proporzionale alla velocità di rotazione. Dalla misura dell'ampiezza della deflessione indotta dalla forza di Coriolis è pertanto possibile determinare la velocità di rotazione [5] [6].

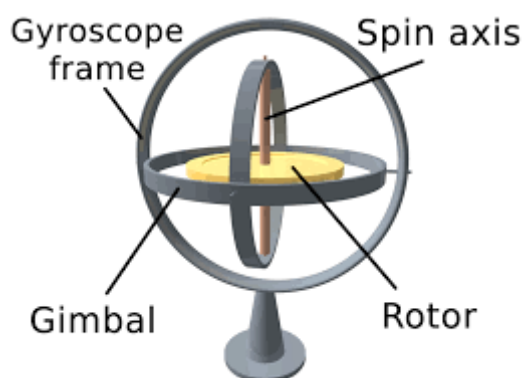


Figura 9 Esempio di giroscopio

3.4.2 SENSORE DI PRESSIONE

Il sensore di pressione utilizzato è di tipo piezoresistivo, questi tipi di sensori di pressione sono costituiti da un diaframma (per la maggior parte in silicene) con estensimetri integrati per rilevare le deformazioni risultanti dalla pressione esercitate su di esso. La configurazione tipica di questi estensimetri è un circuito a ponte di Wheatstone che riduce la sensibilità e aumenta i valori in uscita. Grazie al materiale usato, il limite di pressione è di circa 1000 bar.

Il sensore LPS22HD ha un range di 260 fino a 1260 hPa della pressione assoluta, e presenta una precisione di 0.5 hPa [8].

3.4.3 MAGNETOMETRO 3D

I magnetometri più comuni sono chiamati sensori a effetto Hall a stato solido. Questi sensori utilizzano le proprietà della corrente elettrica che sono influenzate dalla presenza di un campo magnetico che non corre parallelo alla direzione della corrente. Quando c'è un campo magnetico presente, gli elettroni (o il loro opposto, le lacune, o entrambi) nella corrente si accumulano su un lato del materiale conduttivo. Quando è assente, gli elettroni o le lacune si muovono in linea retta. Il modo in cui un campo magnetico influenza il movimento degli elettroni o delle lacune può essere misurato e utilizzato per determinare la direzione di un campo magnetico. I sensori Hall Effect producono anche una tensione proporzionale

all'intensità del campo magnetico, che li rende sia magnetometri vettoriali che scalari.

Il sensore LIS2MDL presenta un range di misura della dinamica del campo magnetico pari a ± 50 gauss, Inoltre presenta 3 canali di misura del campo magnetico [9].

3.4.4 MODULO BLUETOOTH

Il modulo bluetooth SPBTLE-RF permette un facile uso del sistema Bluetooth a bassa energia come master/slave network. Il modulo supporta contemporaneamente diverse operazioni, e può agire allo stesso tempo come un sensore bluetooth a bassa energia e come hub device [10].

3.4.5 COLLEGAMENTI SULLA SCHEDA

La scheda STEVAL-FCU001V1 include diverse connessioni hardware [11]:

- USB: micro B, connettore di tipo femmina
- Connettore a 2-pin per la batteria
- 4 connettori a 2-pin per i motori
- Connettore a 4-pin per UART
- Connettore a 4-pin per I2C
- Connettore a 6-pin per il PWM
- Micro SWD

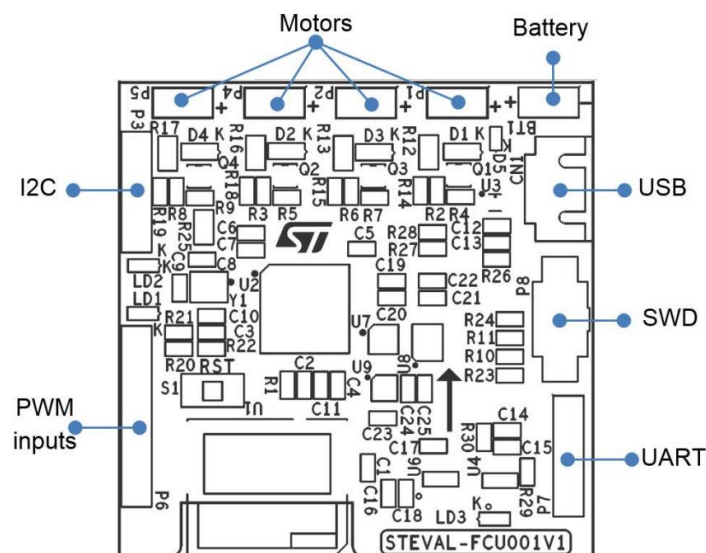


Figura 10 Descrizione delle connessioni

La scheda può essere alimentata tramite il collegamento USB oppure tramite una batteria (a singola cella).

Tabella 1 Connettore a 2-pin per la batteria (BT1)

Pin	Segnale	descrizione
+	VBAT+	Cella batteria di tipo Li-Po da 3.7V
-	GND	

Il verso positivo di collegamento (+) e il piedino di sinistra. È molto importante non sbagliare la polarità di collegamento perché non è presente il diodo di protezione da inversione accidentale della polarità della tensione di alimentazione.

Ognuno dei quattro connettori dei motori è collegato ad un singolo motore alimentato a una tensione di 3.7 V

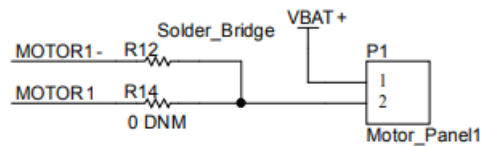
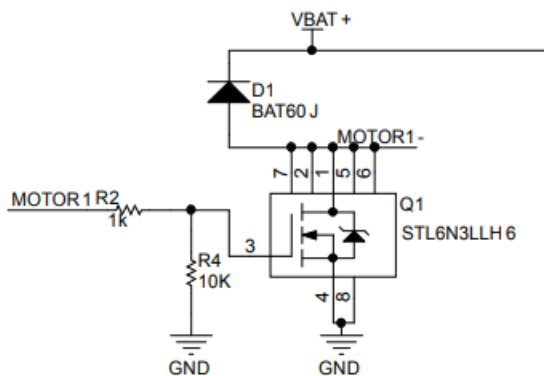


Figura 11 Connettore per il motore M1

Figura 12 Controllo del PWM tramite mosfet per il motore M1

Tabella 2 Connettori a 2-pin per i motori (P1, P2, P4, P5)

Pin	Segnale	descrizione
1	VBAT+	Connesso al polo positivo del motore
2	MOTOR-	Connesso al polo negativo del motore

Come molti controllori di volo, STEVAL-FCU001V1 supporta dei protocolli di comunicazione come l'UART e l'I2C che si interfacciano con i piedini di connesine esterni.

Tabella 3 Connettore a 4-pin per UART (P7)

Pin	Segnale	descrizione
1	VDD	3.3 V di STM32
2	GND	
3	USART1_RX	RXD per STM32
4	USART1_TX	TXD per STM32

Con riferimento alla figura 9, il pin 1 si trova nella parte superiore del collegamento.

Tabella 4 Connettore a 4-pin per I2C (P3)

Pin	Segnale	descrizione
1	VDD	3.3 V di STM32
2	I2C2_SDA	
3	I2C2_SCL	
4	GND	

Con riferimento alla figura 9, il pin 1 si trova nella parte superiore del collegamento.

Il software è costruito per permettere all'utente di controllare il drone tramite app per smartphone oppure tramite un radio comando esterno.

Nel caso si voglia utilizzare un radio comando bisognerà collegare un modulo alla scheda STEVAL-FCU001V1 tramite i connettori P6.

L'implementazione del firmware è compatibile con una ricevente radio con uscite PPM.

Le uscite della ricevente radio vanno collegate ai pin del connettore P6, tenendo conto che (si veda tabella 5):

- CH1 è relativo al controllo del rollio (nel caso di un aereo, controlla la posizione degli alettoni delle ali – in inglese AILERON - AIL)
- CH2 è relativo al controllo del beccheggio (nel caso di un aereo, controlla la posizione degli equilibratori – in inglese ELEVATOR - ELE)
- CH3 è relativa al controllo della spinata (nel caso di un aereo, controlla l'accelerazione del motore – in inglese THROTTLE – THR)
- CH4 è relativa al controllo dell'imbardata (nel caso di un aereo, controlla la posizione del timone – in inglese RUDDER - RUD)

Tabella 5 Connettore a 6-pin per il PWM (P6)

Pin	Segnale	descrizione
1	VBAT+	Direttamente connesso alla batteria (+)
2	TIM2_CH1	
3	TIM2_CH2	
4	TIM2_CH3	
5	TIM2_CH4	
6	GND	

Tabella 6 Debugging micro SWD (P8)

Pin	Segnale	descrizione
1	VDD	
2	SWDD	SWD linea di debugging dei dati
3	GND	
4	SWCLK	SWD linea di debugging dei timer
5	GND	
6	N.C.	
7	GND	
8	N.C.	
9	GND	
10	NRST	Reset per STM32

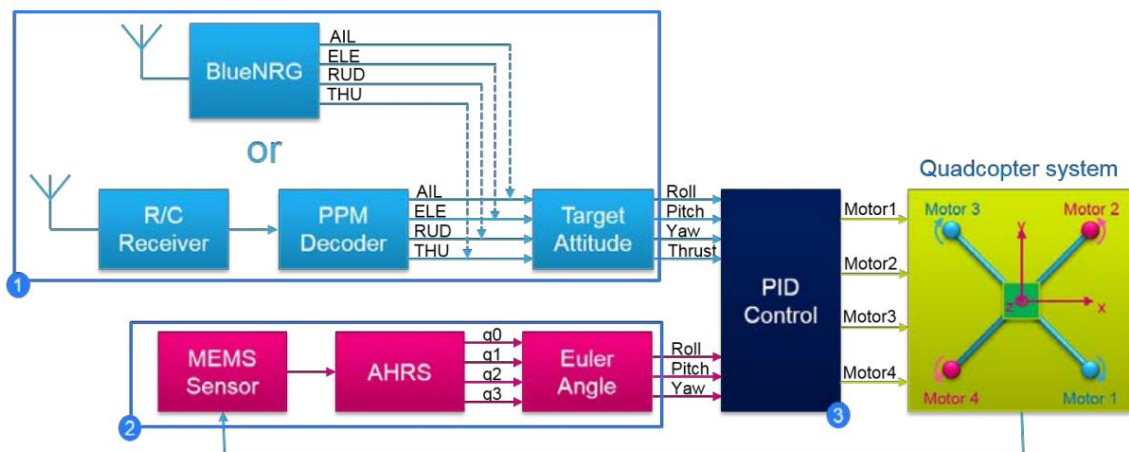
4. FIRMWARE

Il firmware open-source sviluppato da STMicroelectronics è concepito per il controllo di volo di un mini quadricottero ed è strutturato in 6 sezioni principali:

1. un'interfaccia di gestione dei riferimenti per il comando di volo, che può ricevere dei comandi da un dispositivo remoto (radiocomando o app su smartphone);
2. gestore dei dati ricevuti dai sensori;
3. Sistema di riferimento per l'assetto e la direzione, noto come AHRS, utilizza i dati ricevuti dai sensori stimare l'assetto e la direzione (rappresentati mediante quaternioni);
4. controllo di volo che fonde l'uscita dell'AHRS e le informazioni provenienti dal controllo remoto per definire la strategia di volo;
5. controllo di assetto mediante regolatore PID;
6. controllo PID per impartire i comandi ai 4 motori del mini quadricottero.

La progettazione della architettura del firmware è basata su 3 blocchi principali che al loro interno contengono le sei principali funzioni [12].

Figura 13 Architettura del firmware



I 3 blocchi sono organizzati nel modo seguente:

1. Il Blocco è costituito da una prima fase di ricezione dei comandi da un telecomando o da smartphone, da una seconda fase dove tali comandi vengono tradotti in segnali di riferimento da passare al controllore di volo. Il blocco si compone da:
 - Un ricevitore R/C
 - Un decoder PPM
 - Un sistema di traduzione per l'impostazione dell'assetto da seguire
 - Un BlueNRG che permette di ricevere dati tramite una connessione bluetooth

2. Blocco di stima dell'orientazione, composta da:
 - sensori MEMS inerziali per la misura delle accelerazioni lineari (accelerometro) e velocità angolari (giroscopio) lungo/attorno ai 3 assi principali del drone.
 - algoritmo di stima d'assetto e direzione (Attitude & Heading Reference System – AHRS).
 - Euler Angle esegue la conversione dalla notazione utilizzata per esprimere la stima dell'orientazione corrente, passando da quaternioni (utilizzati nell'algoritmo di stima dell'assetto) ad angoli di Eulero (utilizzati nei controlli d'assetto PID).
3. Blocco con all'interno il controllore PID che tramite la differenza tra la stima dell'orientamento corrente e l'assetto impostato fornisce ai motori un certo comando per regolare l'orientazione dell'assetto desiderato.

4.1 RICEVITORE R/C E DECODER PPM

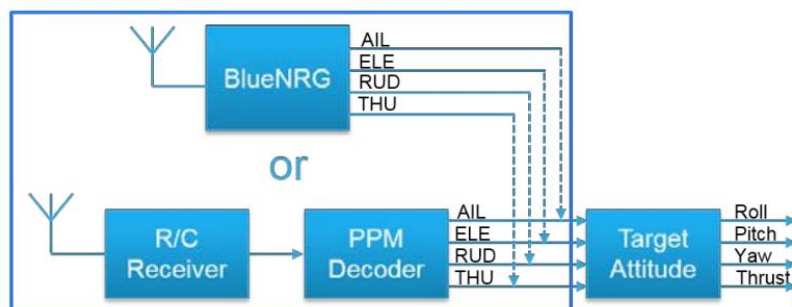


Figura 14 Blocco con all'interno il ricevitore e il decoder PPM

Il modulo di ricezione dei segnali è implementato nel sottoprogramma nominato come **rc.c** e implementa un'interfaccia per lavorare con i segnali trasmessi da un controllo da remoto.

Quando si usa il sistema di controllo tramite uno smartphone si utilizza il dispositivo BlueNRG che permette di comunicare con i dispositivi tramite il bluetooth. Il sistema BlueNRG elabora autonomamente i dati provenienti dal dispositivo di comunicazione usato e permette di salvare i dati risultanti direttamente all'interno delle variabili utilizzate nel codice:

```
72 /* Global R/C data */
73 int16_t gAIL, gELE, gTHR, gRUD;
```

queste variabili rappresentano le 4 direzioni che si possono impostare dal controllo da remoto.

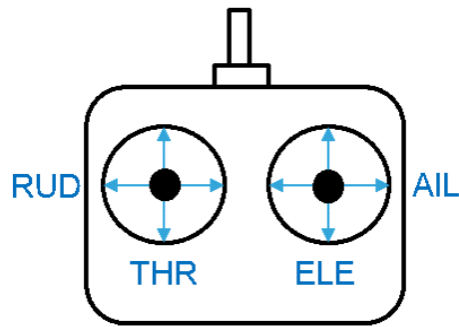


Figura 15 Rappresentazione del telecomando e delle assi di comando

canale	comando	Funzione	Posizione
CH1	AIL	Roll	R:LR
CH2	ELE	Pitch	R:UD
CH3	THR	Thrust	L:UD
CH4	RUD	Yaw	L:LR

Figura 16 Tabella dei comandi del radiocomando

Se si vuole effettuare un cambio del tipo di dispositivo di comando da quello via bluetooth a quello tramite un sistema esterno collegato ai connettori del PWM (P6) si dovrà andare nel file nominato come **rc.h** in cui sono presenti due righe di codice che impostano il tipo di comando che si vuole utilizzare.

```

10 // #define REMOCON_PWM // External Remocon RX 4 channel PWM
11 #define REMOCON_BLE // BLE Remocon App
  
```

Quando si utilizza il sistema esterno il ricevitore dovrà comunicare ai connettori del PWM una serie di dati che dovranno essere elaborati prima di poterli utilizzare.

Un primo processo è quello di input capture, in cui viene calcolato il durata dell'impulso proveniente dal PPM tramite il TIM2 (nei 4 canali).

Con la funzione `void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)` viene elaborato l'input capture dei canali per calcolare la pulsazione di ogni canale del RC e salva il valore al interno della variabile globale `rc_t[4]`.

Chiamando la funzione `update_rc_data(idx)` si potrà convertire la durata dell'impulso dei segnali nel reale segnale di comando immesso nelle variabili globali: `gAIL`, `gELE`, `gTHR`, `gRUD`.

4.1.1 PROCEDURA DI CALIBRAZIONE DEI SENSORI

La calibrazione dei sensori avviene al collegamento della batteria alla scheda. Se il mini quadricottero oppure la scheda in sé non sono su una superficie piana si potrà avviare una procedura di calibrazione tramite il controllo da remoto.

La procedura di calibrazione può essere modificata tramite la funzione `update_rc_data`:

```

219 // Activate Calibration Procedure
220 if ( (gTHR == 0) && (gELE < - RC_CAL_THRESHOLD) && (gAIL > RC_CAL_THRESHOLD) && (gRUD < - RC_CAL_THRESHOLD) )
221 {
222     rc_cal_flag = 1;
223 }

```

In tale procedura si dovrà seguire una serie di comandi con il controllo da remoto, in cui si dovrà stare in una posizione di standby e poi spostare gli analogici in una certa posizione, come illustrano in figura 16:

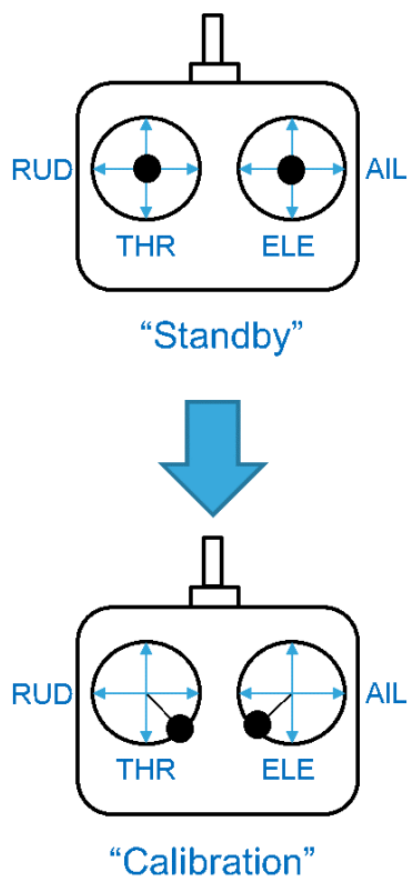


Figura 17 Sequenza di attivazione della procedura di calibrazione dei sensori

I comandi che danno l'input di inizio alla calibrazione, come si è visto, sono all'interno del file `rc.c` però la procedura di calibrazione in sé viene eseguita all'interno del file `main`. In cui prima acquisisce i dati provenienti dai sensori di accelerazione e giroscopio e in seguito li salva nella variabile che indica gli offset dei sensori.

4.1.2 PROCEDURA DI ABILITAZIONE DEI MOTORI

Per evitare possibili danneggiamenti o infortuni quando la batteria viene collegata e il controllo da remoto viene attivato, si è pensato di inserire una procedura di abilitazione (arming) dei motori: i motori rimarranno sempre disattivati fin quando non si eseguirà una procedura simile a quella di calibrazione dei sensori.

La procedura può essere modificata nuovamente tramite la funzione `update_rc_data`:

```
225 // Activate Arming/Disarming
226 if ( (gTHR == 0) && (gELE < - RC_CAL_THRESHOLD) && (gAIL < - RC_CAL_THRESHOLD) && (gRUD > RC_CAL_THRESHOLD))
227 {
228     if (rc_enable_motor==0) // if not armed -> arm
229     {
230         rc_enable_motor = 1;
231         fly_ready = 1;
232     }
233     else // if armed -> disarm
234     {
235         rc_enable_motor = 0;
236         fly_ready = 0;
237     }
238 }
239 }
```

Per armare e disarmare bisognerà eseguire una nuova sequenza di comandi dal sistema di controllo da remoto:

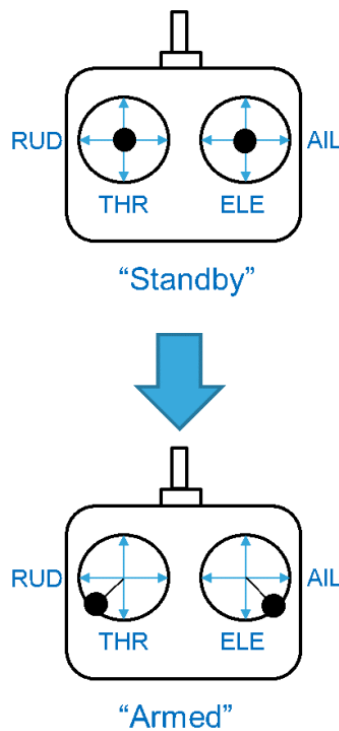


Figura 18 Sequenza di attivazione (arming) dei motori

4.1.3 CALIBRAZIONE E ABILITAZIONE NEL SISTEMA BLUENRG

Utilizzando come sistema di controllo da remoto lo smartphone la procedura precedentemente illustrata viene notevolmente semplificata, essendo che sono disponibili degli appositi pulsanti per la calibrazione dei sensori e l'arming dei motori all'interno dell'app.

Una volta fatto il tap sul bottone si avvierà la procedura selezionata e dopo pochi secondi il bottone si colorerà di verde per segnalare che la procedura è stata eseguita con successo.



Figura 19 Rappresentazione della schermata di comando tramite smartphone

I vari bottoni sono:

1. Bottone per il collegamento al drone tramite il bluetooth
2. Bottone di calibrazione dei sensori
3. Bottone per armare i motori

4.2 TARGET ATTITUDE

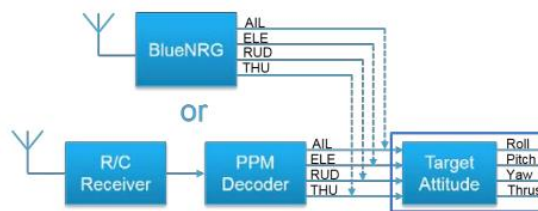


Figura 20 Modulo di Target Attitude

Il modulo di *Target Attitude* ha lo scopo di convertire i comandi provenienti dal controllo remoto in dei riferimenti che possono essere poi confrontati con le stime fornite dal modulo AHRS. Il modulo è implementato nel file **rc.c** con la funzione `void GetTargetEulerAngle(EulerAngleTypeDef *euler_rc, EulerAngleTypeDef *euler_ahrs)`.

I dati provenienti dal telecomando (per esempio gELE) vengono normalizzati in una scala per poi poter essere convertiti in dei angoli:

```

246     t1 = gELE;
247     if (t1 > RC_FULLSCALE)
248         t1 = RC_FULLSCALE;
249     else if (t1 < -RC_FULLSCALE)
250         t1 = - RC_FULLSCALE;
251     euler_rc->thx = -t1 * max_pitch_rad / RC_FULLSCALE;

```

In questo esempio si può notare come gELE viene trasformato in un movimento angolare di pitch, per cui se tramite il telecomando viene inviato un comando di gELE positivo (ciò rappresenta la volontà di far muovere il mini quadricottero in avanti) si dovrà creare un movimento per cui l'angolo di pitch risulti negativo così che alzando il lato inferiore del mini quadricottero esso sia portato ad eseguire un avanzamento, per questo motivo nella funzione di comando è presente la variabile (-t1).

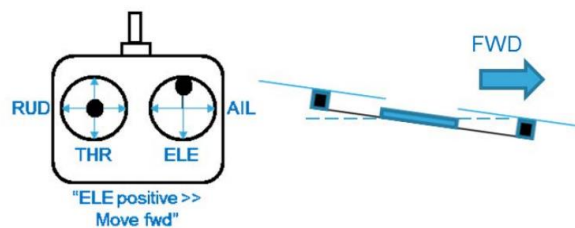


Figura 21 Rappresentazione del comando d'avanzamento

La stessa cosa succede anche per il movimento di gAIL che sarà trasformato in un movimento angolare di roll per cui se si vorrà far muovere il mini quadricottero verso destra si dovrà creare un angolo di roll che porterà il lato sinistro ad alzarsi.

Per quanto riguarda invece le variabili gRUD e gTHR non hanno bisogno di ulteriori modifiche, oltre a quello di adattamento di scala perché i loro movimenti sono in linea con quelli forniti dal telecomando.

4.3 MEMS SENSOR



Figura 22 Modulo contenente i dati dei sensori

Nel modulo MEMS Sensor sono presenti 2 sensori di tipo inerziale accelerometro e giroscopio 3D, essi invieranno i dati acquisiti al modulo AHRS che stimerà l'assetto (orientazione) del drone.

Il modulo converte i dati provenienti dai sensori montati sul mini quadricottero nelle relative coordinate; le assi X e Y. Le coordinate sono disposte in linea con i sensori di accelerazione e giroscopio.

I dati iniziali provenienti dalle 6 assi dell'accelerometro e del giroscopio sono convertite in [mg] (rappresenta la forza misurata in confronto con la forza di gravità) e [mdps] (gradi °/ secondo) vengono trasmessi al sistema di coordinate.

4.4 ATTITUDE & HEADING REFERENCE SYSTEM (AHRS)

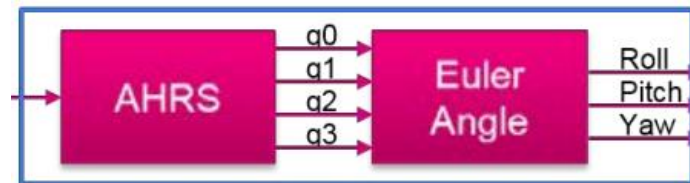


Figura 23 modulo AHRS

L'AHRS (Attitude and Heading Reference System) è un algoritmo che utilizzando i dati provenienti dai sensori di accelerazione e giroscopio riesce ad ottenere una stima dell'orientazione tramite gli angoli di pitch, roll e yaw.

In genere l'AHRS utilizza inoltre un magnetometro che permette di stimare la direzione di avanzamento (angolo di heading, rispetto al polo nord magnetico) nel calcolo della posizione, in questo caso non viene usato all'interno dell'algoritmo perché è stato deciso di usare l'algoritmo basato sul metodo di Mahoney [13][14].

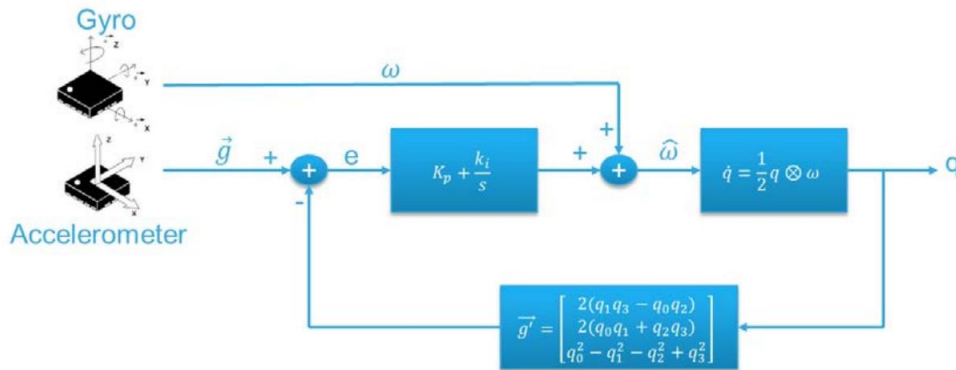


Figura 24 Algoritmo di Mahoney

Per quanto riguarda l'implementazione nel codice esistono due file uno denominato **quaternion.c** e l'altro **ahrs.c**. il primo è un file dove sono inserite le principali e più semplici operazioni svolte con i quaternioni che permettono di ottenere i riferimenti degli angoli calcolati, nel secondo file sono presenti tutte le altre operazioni necessarie per combinare i dati ottenuti dall'accelerometro e giroscopio per ottenere la stima dell'assetto.

I dati grezzi del accelerometro e giroscopio sono trasferiti all'algorithm AHRS nel file **ahrs.c** tramite la funzione `ahrs_fusion_ag(&acc_ahrs, &gyro_ahrs, &ahrs);`; la trasformazione dei dati del AHRS nei angoli necessari (Euler angle) avviene nel file **quaternion.c** per mezzo della funzione `QuaternionToEuler(&ahrs.q, &euler_ahrs);`.

4.5 CONTROLLORE DI VOLO CON PID

L'assetto del mini quadricottero viene stabilizzato tramite controllo PID (Proporzionale, Integrale, Derivativo). Il controllore PID riceve in ingresso l'errore di assetto determinato come differenza tra il riferimento d'assetto inviato dal telecomando, e la stima dell'assetto ottenuta dall'AHRS.

L'uscita del sistema PID è determinata dalla somma delle tre azioni: proporzionale, integrale, derivativa.

Azione proporzionale

Nel caso di un quadricottero, la componente proporzionale è fondamentale perché controlla l'errore istantaneo.

Più alto è il coefficiente di proporzionalità maggiore risulterà la sensibilità e la reattività a modifiche della sua posizione. Viceversa, se è troppo basso il quadricottero risulterà lento nel cambiamento di assetto e sarà difficile da manovrare.

Azione integrale

L'azione integrale fa sì che il controllore abbia memoria dei valori del segnale di errore nel tempo; in particolare, il valore dell'azione integrale non è necessariamente nullo se è nullo l'errore. Questa proprietà dà al PID la capacità di portare il processo esattamente al punto di riferimento richiesto, dove la sola azione proporzionale risulterebbe nulla.

Il problema più comune legato alla componente integrativa è il cosiddetto windup. L'insorgenza del windup può verificarsi quando il comando di attuazione è vincolato a dei valori massimi e minimi; in tal caso, supponendo di applicare al sistema di riferimento un disturbo a gradino, l'integratore inizierà ad accrescere il suo valore in uscita, per via dell'errore non nullo. L'output del controllore è allora tale da raggiungere il valore massimo del comando e questo rimarrà costante, anche se l'uscita dell'integratore crescerà fino a quando l'errore non sarà nullo.

La soluzione adottata nel firmware è quella di limitare il segnale di controllo al valore limite eseguibile. In questo modo si elimina completamente l'effetto di windup.

Azione derivativa

La componente derivativa compensa rapidamente le variazioni del segnale di errore: se l'errore aumenta, l'azione derivativa cerca di "frenare" questa deviazione, in ragione della sua velocità di cambiamento, senza aspettare che l'errore diventi significativo (azione proporzionale) o che persista per un certo tempo (azione integrale).

Il termine derivativo permette al drone di raggiungere più velocemente l'angolo desiderato.

4.5.1 IMPLEMENTAZIONE DEL PID NEL FIRMWARE

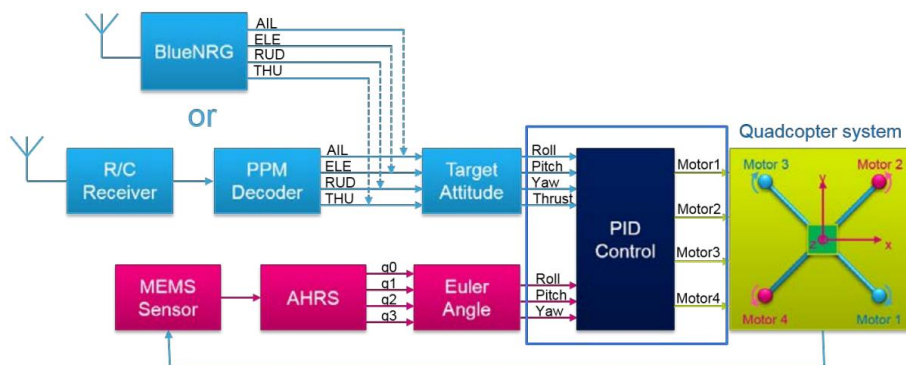


Figura 25 Blocco di controllo PID

Il controllo tramite PID è implementato nel file **flight_control.c**.

Si tratta di una implementazione in cascata con due anelli di controllo con regolatore PID:

1. Anello di controllo esterno (vedi figura 24): compara gli angoli derivanti dal sistema AHRS e gli angoli provenienti dal telecomando, controlla l'inclinazione angolare tramite la funzione:
`FlightControlPID_OuterLoop(&euler_rc_fil, &euler_ahrs, &ahrs, &pid);`
2. Anello di controllo interno (vedi figura 24): insegue il riferimento di velocità angolare fornito dal controllore PID dell'anello esterno tramite la funzione:
`FlightControlPID_innerLoop(&euler_rc_fil, &gyro_rad, &ahrs, &pid, &motor_pwm);`

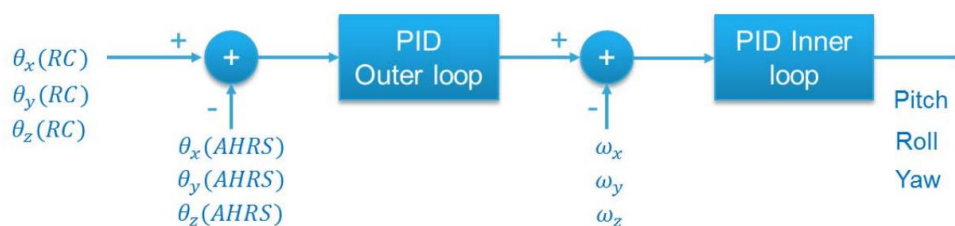


Figura 26 Controllo PID in cascata: anello esterno e anello interno

L'output del PID viene convertito in un comando di tensione (duty-cycle del segnale PWM) da fornire ai 4 motori del mini quadricottero (vedi anche figura 25) :

```

266 motor_pwm->motor1_pwm = motor_thr - pid->x_s2 - pid->y_s2 + pid->z_s2 + MOTOR_OFF1;
267 motor_pwm->motor2_pwm = motor_thr + pid->x_s2 - pid->y_s2 - pid->z_s2 + MOTOR_OFF2;
268 motor_pwm->motor3_pwm = motor_thr + pid->x_s2 + pid->y_s2 + pid->z_s2 + MOTOR_OFF3;
269 motor_pwm->motor4_pwm = motor_thr - pid->x_s2 + pid->y_s2 - pid->z_s2 + MOTOR_OFF4;

```

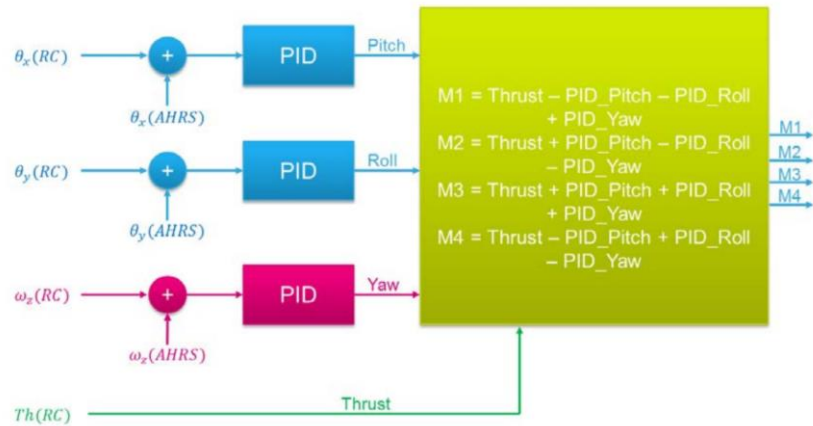


Figura 27 Uscita del controllo PID convertito nel comando di tensione PWM

4.6 FILE main.c

Il file di main integra tutte le funzioni precedentemente riportate richiamandole dai file in cui sono state costruite.

Il main riporta dunque le principali operazioni che saranno eseguite all'interno del firmware, il flusso delle operazioni eseguite può essere rappresentato come in figura 27:

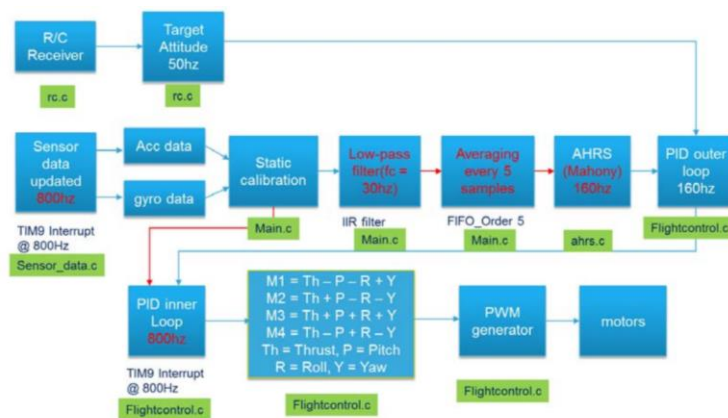


Figura 28 rappresentazione del flusso di comandi nel file main.c

Configurazione del MCU

Nella sezione di configurazione del MCU vengono effettuate le prime operazioni di configurazione dei parametri e le inizializzazioni necessarie per far funzionare correttamente il microcontrollore.

In particolare, vengono richiamati tutti i file che presentano l'inizializzazione delle varie funzioni che sono state spiegate precedentemente.

Inizializzazione delle periferiche

La configurazione delle periferiche è essenziale per farle interagire correttamente con il firmware.

All'interno del firmware studiato sono presenti le periferiche:

- GPIO (General Purpose Input/Output)
- ADC (Analog Digital Converter)
- TIM2 (timer predisposto per il controllo del PWM input)
- TIM4 (timer predisposto per il controllo del PWM dei motori)
- TIM9 (timer predisposto per il controllo delle funzioni interne al firmware)
- UART (Universal Asynchronous Receiver-Transmitter)

Ciclo di controllo

Il ciclo di controllo non viene eseguito per intero nel main ma viene affidato ai sottoprogrammi studiati precedentemente per eseguire le funzioni più complesse.

Principalmente nel file di main vengono impostate le frequenze a cui vengono richiamate le principali funzioni che compongono il firmware. La lettura dei sensori viene eseguita mediante interrupt ad una frequenza di 800 Hz, la funzione per l'esecuzione del anello esterno viene richiamata con un interrupt ad una frequenza di 160 Hz a differenza dell'anello interno che viene richiamato ad una frequenza di 800 Hz.

5. CONCLUSIONI

In conclusione, possiamo dire che il mini quadricottero supportato dalla scheda elettronica STEVAL-FCU001V1 è un facile primo approccio al mondo dei quadricotteri con una grande possibilità di modifiche future.

La facilità deriva essenzialmente dal fatto che i componenti sono facilmente reperibili e il firmware è disponibile gratuitamente su GitHub. In oltre il firmware è stato studiato appositamente per essere ulteriormente modificato e di facile comprensione, i commenti al suo interno sono molti e aiutano nella comprensione del file che si sta visionando.

Alcune possibilità di modifiche future potrebbe essere quella di aumentare le dimensioni del quadricottero in modo tale da poter ospitare una maggiore quantità di sensori e di poter installare anche una fotocamera. Un ulteriore modifica futura potrebbe essere quella di elaborare il firmware con il software matlab.

BIBLIOGRAFIA

- [1] Wikipedia, "Aeromobile a pilotaggio remoto".
https://it.wikipedia.org/wiki/Aeromobile_a_pilotaggio_remoto#:~:text=Un%20aeromobile%20a%20pilotaggio%20remoto,un%20pilota%20umano%20a%20bordo.
- [2] STMicroelectronics, UM2311. *Getting started with the STEVAL-FCU001V1 flight controller unit evaluation board for mini drones*, pp. 2.
- [3] Eurek. "Microcontrollori: Cosa sono e a cosa servono nell'elettronica?".
<https://eurek.it/microcontrollori-cosa-sono-e-a-cosa-servono-nell-elettronica/>
- [4] STMicroelectronics, STM32F401
<https://www.st.com/en/microcontrollers-microprocessors/stm32f401.html>
- [5] R. Oboe, R. Antonello, E. Lasalandra, G. S. Durante and L. Prandi, "Control of a Z-axis MEMS vibrational gyroscope," in IEEE/ASME Transactions on Mechatronics, vol. 10, no. 4, pp. 364-370, Aug. 2005
- [6] R. Antonello, R. Oboe, L. Prandi and F. Biganzoli, "Automatic Mode Matching in MEMS Vibrating Gyroscopes Using Extremum-Seeking Control," in IEEE Transactions on Industrial Electronics, vol. 56, no. 10, pp. 3880-3891, Oct. 2009
- [7] STMicroelectronics. LSM6DSL, iNEMO 6DoF inertial measurement unit (IMU), for smart phones and battery operated IoT, Gaming, Wearable and Consumer Electronics. Ultra-low power and high accuracy.
<https://www.st.com/en/mems-and-sensors/lsm6dsl.html>
- [8] STMicroelectronics. LPS22HH, High-performance MEMS nano pressure sensor: 260-1260 hPa absolute digital output barometer.
<https://www.st.com/en/mems-and-sensors/lps22hh.html>
- [9] STMicroelectronics. LIS2MDL, Magnetic sensor, digital output, 50 gauss magnetic field dynamic range, ultra-low power high performance 3-axis magnetometer.
<https://www.st.com/en/mems-and-sensors/lis2mdl.html>
- [10] STMicroelectronics. SPBTLE-RF, Very low power network processor module for Bluetooth® low energy v4.1

- [11] STMicroelectronics, user manual: UM2311. *Getting started with the STEVAL-FCU001V1 flight controller unit evaluation board for mini drones*, pp. 5-8.
- [12] STMicroelectronics, user manual: UM2311. *Getting started with the STSW-FCU001 reference design firmware for mini drones*.
- [13] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear Complementary Filters on the Special Orthogonal Group. *IEEE Transactions on Automatic Control*, Institute of Electrical and Electronics Engineers, 2008, 53 (5), pp.1203-1217.
- [14] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Complementary filter design on the special orthogonal group $SO(3)$. *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*. Seville, Spain. December 12-15, 2005.