

Application: Droni con STM32

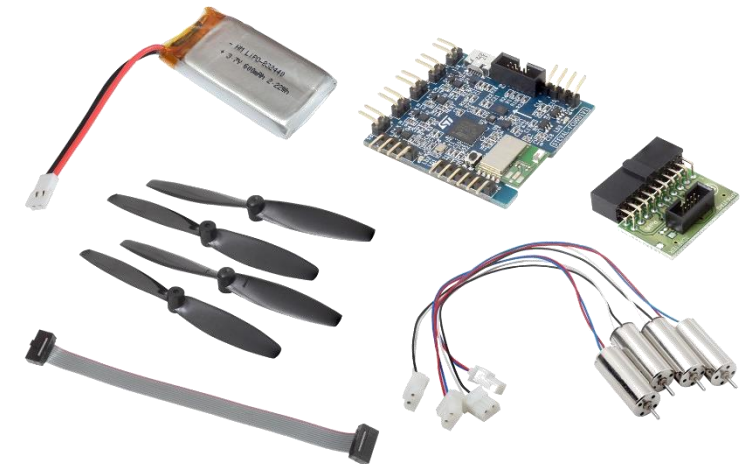
Giuseppe Messina



STEVAL-DRONE01

2

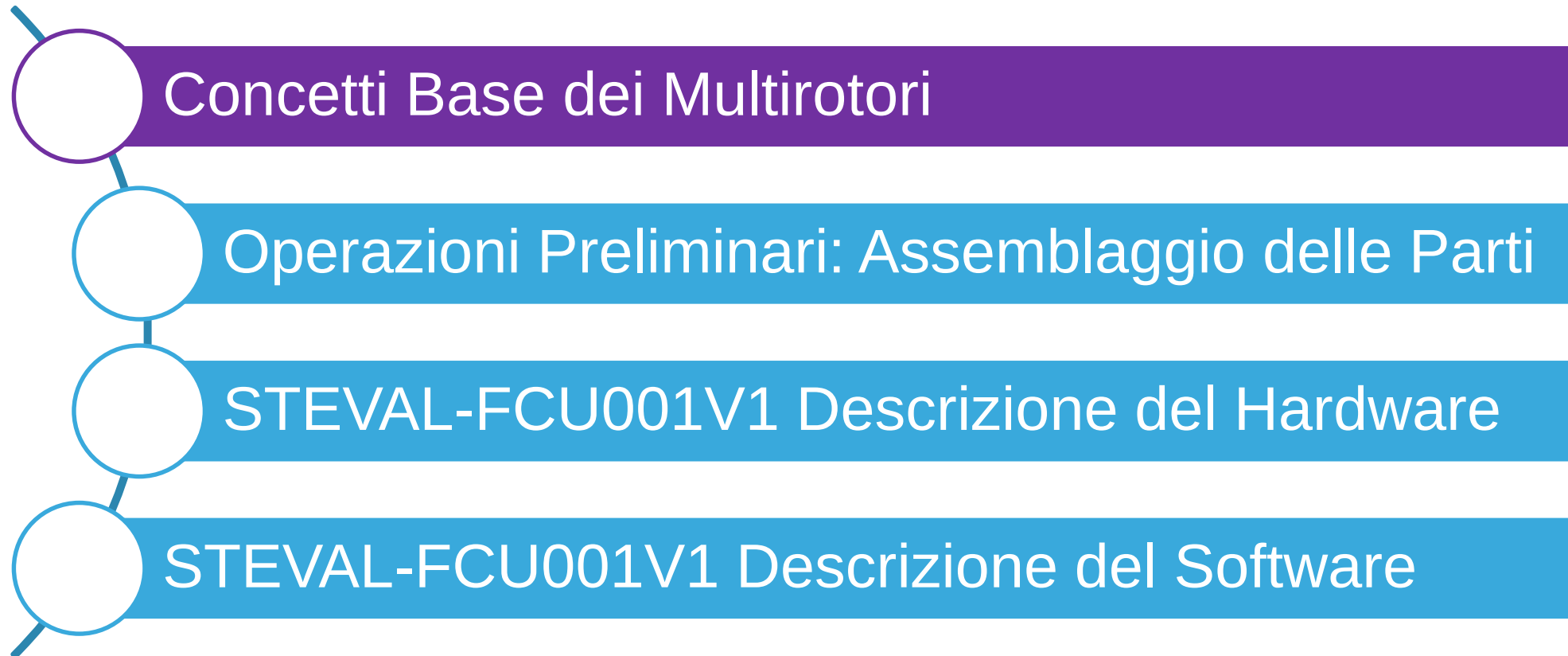
- Nel ambito di questo progetto ST ha deciso per la prima volta di rilasciare un prodotto completo per la valutazione della board **STEVAL-FCU01v1**.
- Il kit è disponibile presso i più famosi distributori di componenti elettronici
- Prezzo tra 45-50 €
<http://bit.ly/2FvIXBM>



Avvertenze Prima del Volo

3

- Questa presentazione ha lo scopo di fornire alcune informazioni di base sui multirotori e il loro controllo di volo, al fine di poter utilizzare la board di riferimento **STEVAL-FCU001V1** ed eventualmente modificarla per le proprie esigenze.
- Vi preghiamo di prestare attenzione al fatto che far volare un **multirottore potrebbe essere pericoloso**, sia per il pilota che per altre persone nelle vicinanze, se le regole di base non vengono rispettate.
- Inoltre, vi preghiamo di rispettare le regole specificate dall'Autorità per l'aviazione civile prima di un volo e di **evitare le "no-fly" zone**.
- Il Firmware fornito da STMicroelectronics è rilasciato al fine di sviluppo e di testing della board STEVAL-FCU001V1, non è garantito privo di bugs e deve essere utilizzato come base di riferimento per lo sviluppo di soluzioni più performanti e sicure. **STMicroelectronics non si assume nessuna responsabilità per uso diversi dal testing.**



Nozioni di base: la struttura meccanica

5



- **Telaio:** ci sono diversi tipi di telai disponibili sul mercato, di diverse dimensioni e materiali. Per l'algoritmo di controllo basilare (FW di riferimento disponibile con STEVAL-FCU001V1) le eliche e i motori devono essere posizionati **equidistanti**.



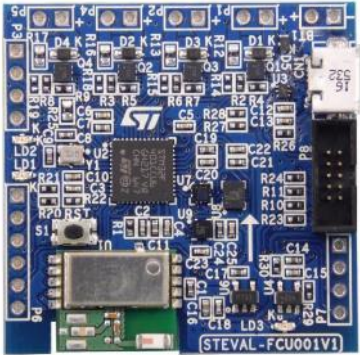
- **Motori:** solitamente per i droni piccoli vengono utilizzati i motori DC a spazzole, mentre per i telai più grandi sono preferibili quelli DC brushless a 3 fasi. La scelta del motore deve essere eseguita insieme al telaio e alle eliche selezionati. Si noti che **alcuni motori sono progettati per essere utilizzati solo in direzione oraria o antioraria** (in particolare i Motori DC a spazzole).



- **Eliche:** è un tipo di ventola che converte il movimento rotatorio in spinta. C'è una vasta scelta con 2, 3 o 4 lame per elica, diversi materiali e diverse forme.

Nozioni di base: l'elettronica

6



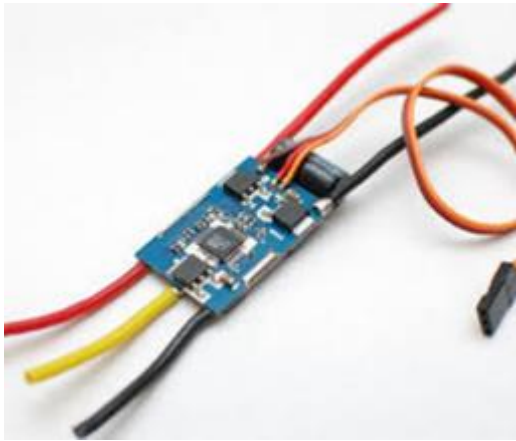
- **FCU (Flight Control Unit):** l'algoritmo principale per il controllo del volo, è solitamente in esecuzione su un microcontrollore a 32 bit, che controlla anche la velocità di ciascun motore, o controllandolo direttamente (Motore DC) o inviando un segnale a un ESC (Electronic Speed Control) nel caso di motori brushless. Al fine di stabilizzare il drone, vengono analizzati i dati di almeno un **accelerometro** e di un **giroscopio**, mentre per il controllo dell'altitudine può essere utilizzato un sensore di pressione.



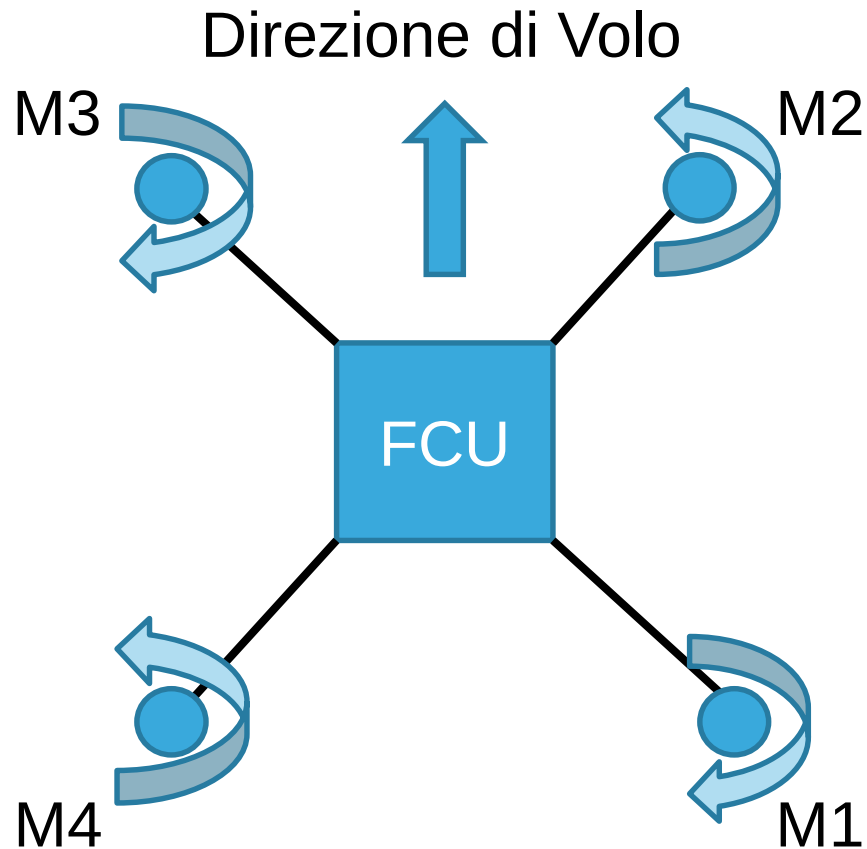
- **Controllo Remoto e Ricevitore:** anche se è possibile usare i droni da smartphone (tramite **Bluetooth o Wifi**), il controllo remoto viene spesso utilizzato per la migliore sensibilità di controllo del volo. Ogni telecomando ha un'unità ricevitore che deve essere collegata alla FCU, e sono spesso venduti accoppiati.



- **Batteria:** per i multicotteri sono comunemente usate batterie LIPO (**Polimeri di Litio**). A seconda delle dimensioni del drone e dei Motori, vengono selezionate batterie da **1, 2 o 3 celle**, con diverse capacità di corrente e dimensioni. Questo tipo di batteria deve essere usato con **particolare attenzione** perché è possibile danneggiarlo (farlo **esplodere!**) se sovraccaricato o cortocircuitato.



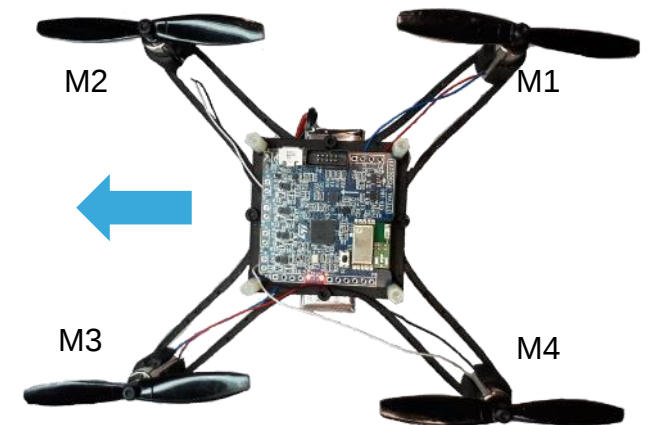
- **ESC (Electronic Speed Control):** è utilizzato per pilotare un motore brushless DC a 3-fasi. Generalmente riceve in ingresso un segnale PWM fino a 500 Hz dalla FCU, con una larghezza di banda che varia da 1 msec (spegnimento del motore) a 2 msec (velocità massima del motore). L'ESC è collegato direttamente alla batteria, mentre la FCU utilizzerà il Vbus per pilotare i motori. La maggior parte di essi incorpora un cosiddetto **BEC (Battery Eliminator Circuit)**, fondamentalmente un convertitore DCDC per fornire la corretta tensione di alimentazione alla FCU collegata.



FCU: Flight Control Unit

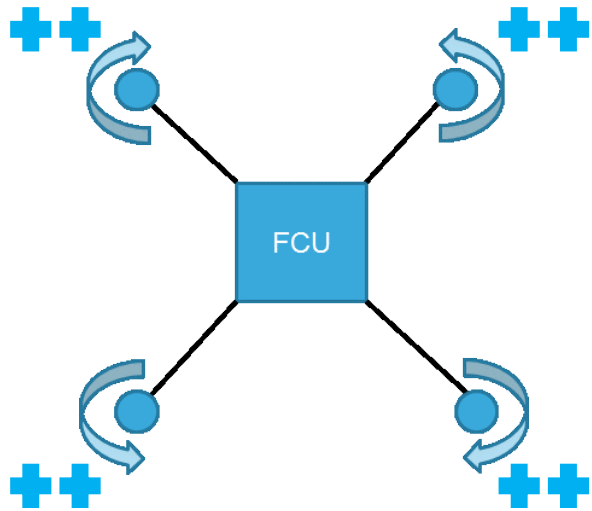
- Ogni motore produce sia una spinta che una coppia attorno al suo centro di rotazione.
- Se tutti i motori girano alla stessa velocità angolare, con R1 e R3 in senso orario e R2 e R4 in senso antiorario, l'accelerazione angolare attorno all'asse di imbardata (YAW) è esattamente zero (quindi **non è necessario un motore di coda come sugli elicotteri**).

La scheda STEVAL-FCU001V1 montata sul frame del drone segue esattamente la stessa convenzione.

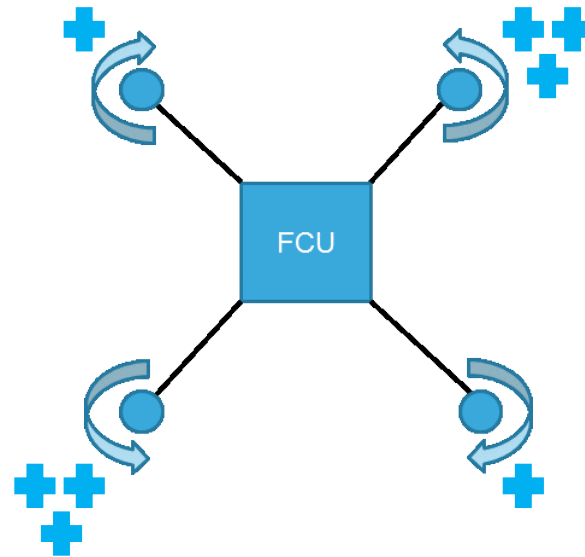


Nozioni di base: Dinamiche di Controllo del volo

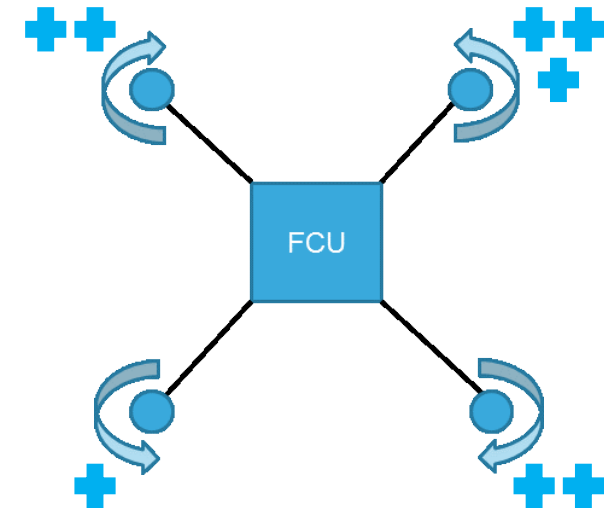
9



Un quadricottero *aleggia o regola la sua altitudine* applicando una spinta uguale a tutti e 4 i motori.



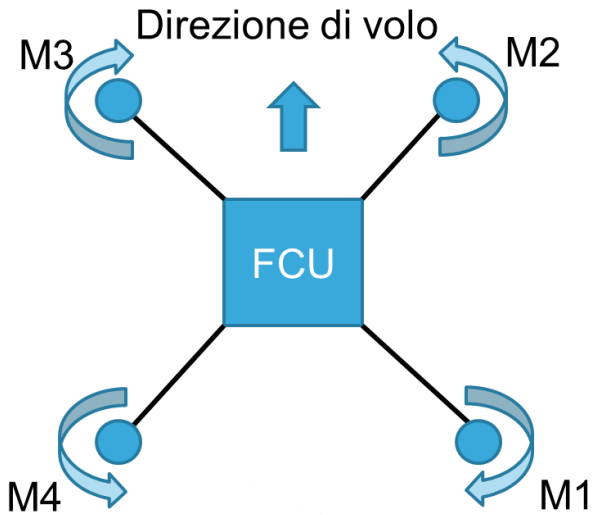
Un quadricottero *regola la sua imbardata* applicando più spinta ai motori che ruotano in una direzione.



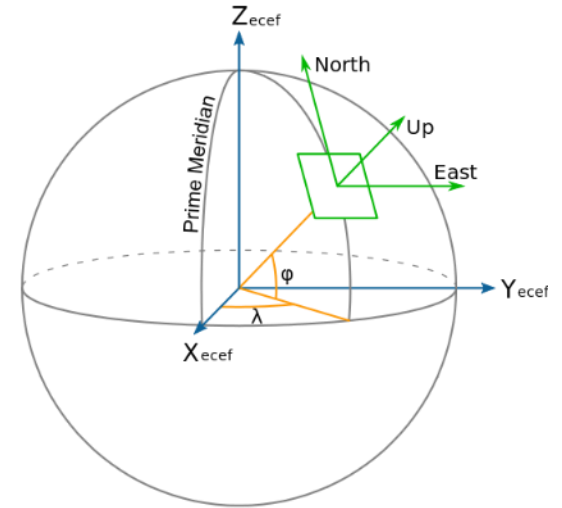
Un quadricottero regola *il beccheggio o il rollio* applicando più spinta a un motore e meno spinta al suo motore diametralmente opposto.

Nozioni di base: Dinamiche di Controllo del volo

10



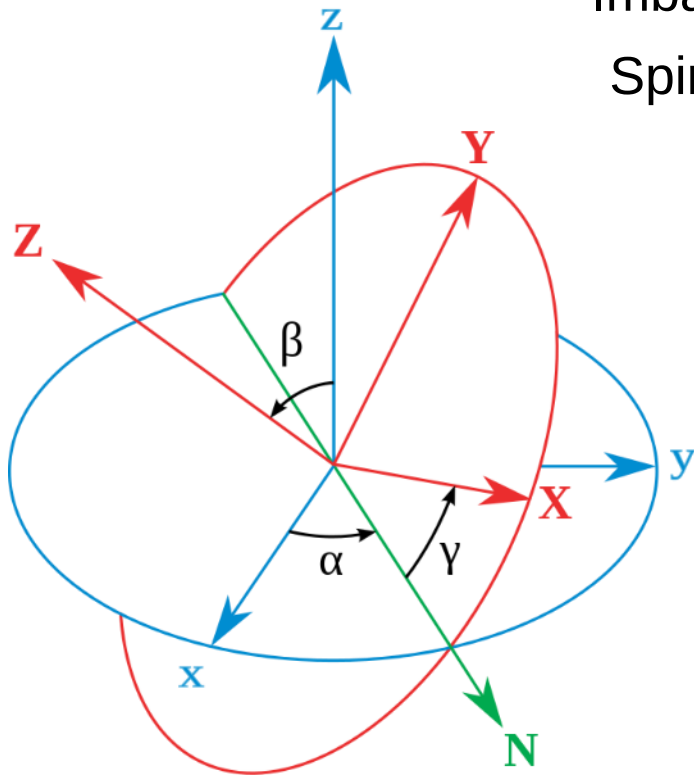
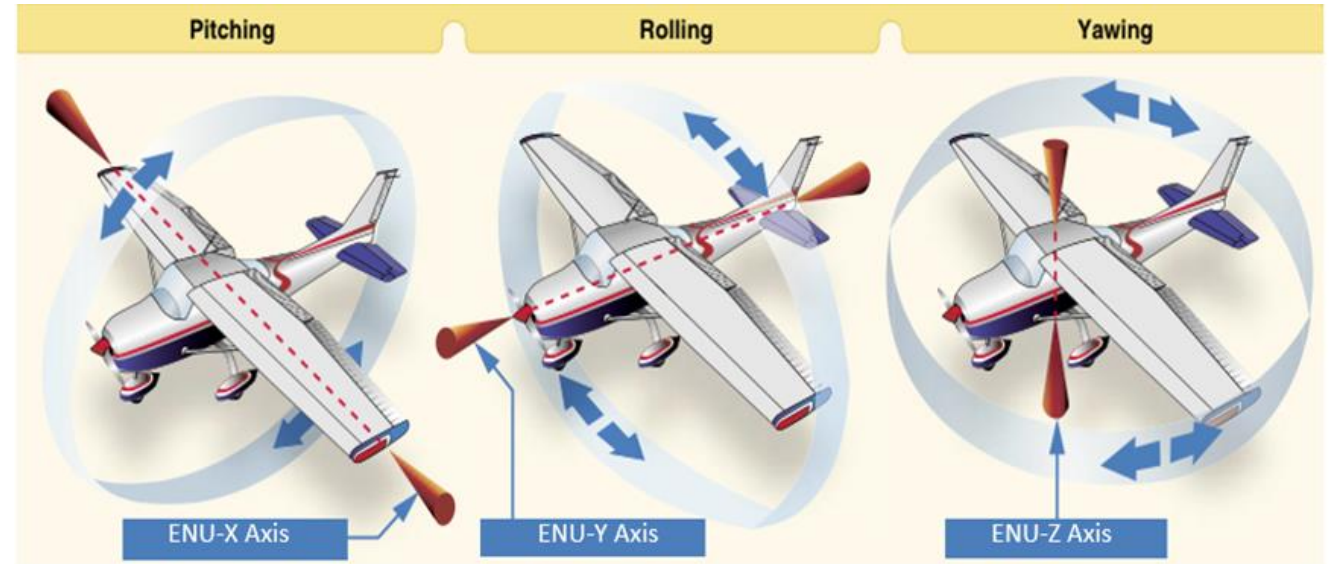
- Sistema di Coordinate : ENU
- Tipo di Telaio: X-type
 - 4 x Motori, con il Sistema di rotazione descritto
- Movimenti:
 - Avanti / Indietro (Pitch)
 - Sinistra / Destra (Roll)
 - Rotazione a Sinistra / Destra (Yaw)
 - Su / Giù



Sistema di Coordinate : ENU

Motore Azione	R1	R2	R3	R4	Note
Pitch (x)	Lento (Veloce)	Veloce (Lento)	Veloce (Lento)	Lento (Veloce)	Movimento Indietro (Avanti)
Roll (y)	Lento (Veloce)	Lento (Veloce)	Veloce (Lento)	Veloce (Lento)	Movimento a Destra (Sinistra)
Yaw (z)	Veloce (Lento)	Lento (Veloce)	Veloce (Lento)	Lento (Veloce)	Rotazione a Sinistra (Destra)
Su / Giù	Veloce (Lento)	Veloce (Lento)	Veloce (Lento)	Veloce (Lento)	Su (Giù)

Rollo - Rolling
Beccheggio - Pitching
Imbardata - Yawing
Spinta - Thrusting



Angoli di Eulero - Il sistema fisso (xyz) è rappresentato in blu, e il sistema ruotato (XYZ) è rappresentato in rosso. La linea dei nodi, indicata con *N*, è rappresentata in verde.

Gli angoli di Eulero sono:

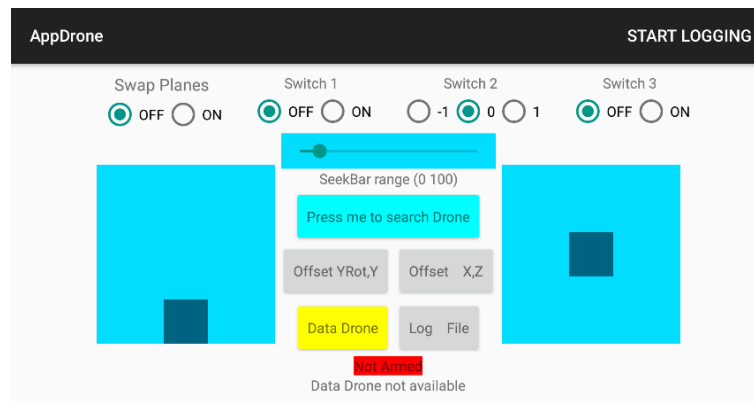
- α è l'angolo tra l'asse *x* e la linea dei nodi *N*.
- β è l'angolo tra gli assi *z* e *Z*.
- γ è l'angolo tra la linea dei nodi *N* e l'asse *X*.

Nozioni di Base: Interfaccia App Android

12



Modulo Bluetooth Low Energy montato sulla STEVAL-FCU001V1

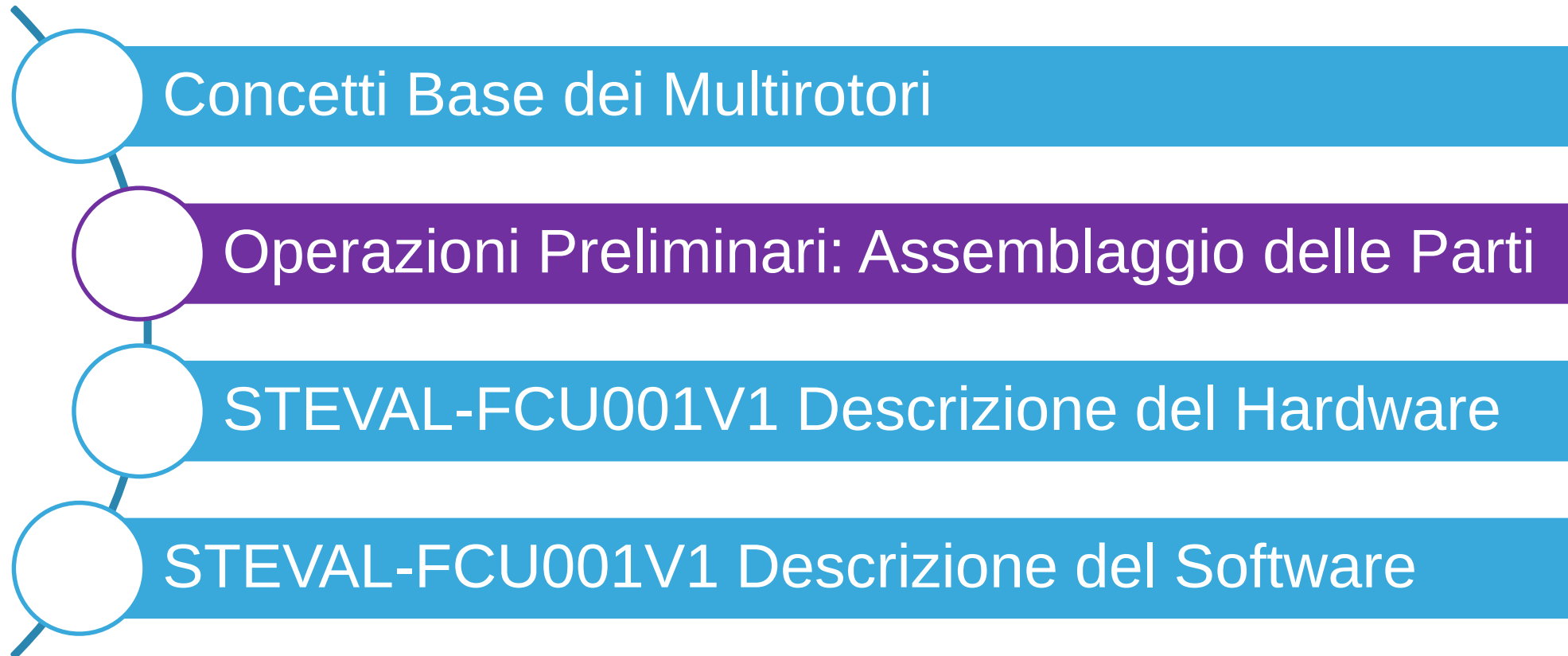


App Android con connessione BLE con layout simile ai telecomandi meccanici generalmente utilizzati.



Controllo Remoto TX e relativo modulo RX (da connettere alla FCU) proprietario.

Attenzione che l'App Android e' attualmente in versione Beta e puo' subire delle modifiche anche rilevanti nelle prossime release.



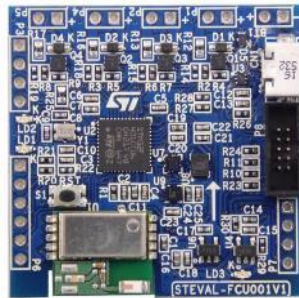
Assemblaggio delle Parti

14

- Gli articoli qui sotto rappresentano solo un esempio di possibile assemblaggio per un piccolo drone giocattolo ed è già stato testato con la versione FW corrente con prestazioni di volo stabili.
- **Attenzione:** scegliendo componenti diversi potrebbe essere necessaria una regolazione FW e/o una messa a punto.



Telaio
stampato 3D



STEVALFCU001v1



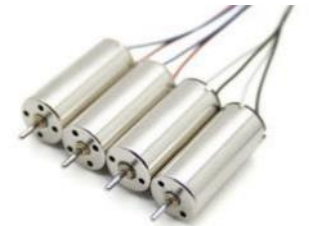
Controllo Remoto e modulo RX



Eliche da 65mm



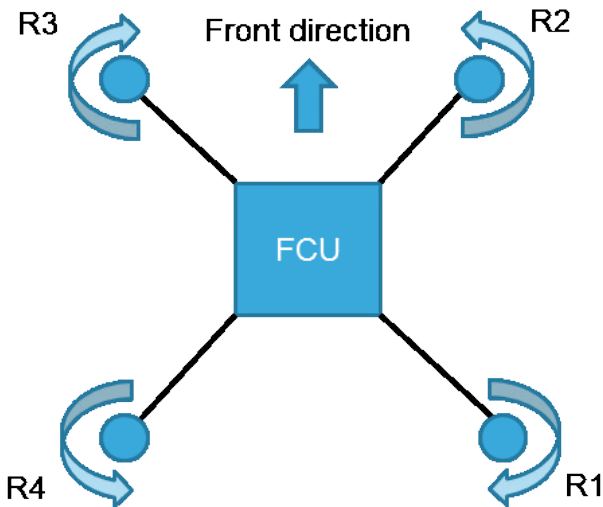
batteria 1-cell
4.2V 500mA



2 x CW, 2 x CCW
Motori brushed DC
8.5x20mm

Passo 1: montaggio della STEVAL-FCU001V1 sul telaio

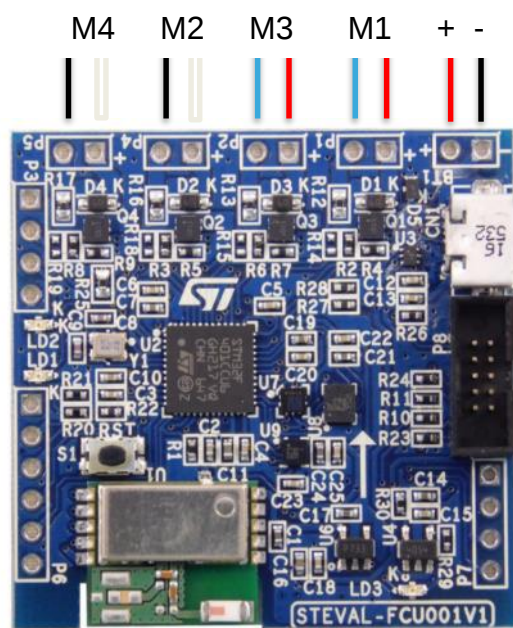
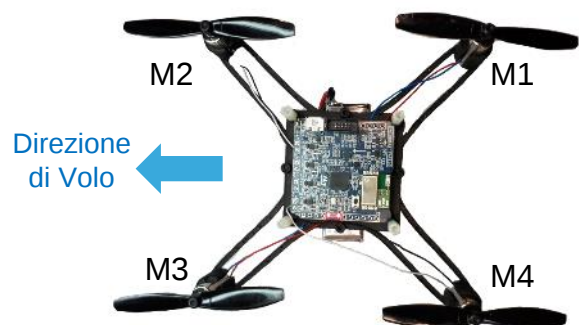
15



- La FCU deve essere montata al centro del drone, equidistante dai quattro motori.
- La direzione di volo è indicata da una freccia sulla STEVAL-FCU001V1.
- La FCU deve essere montata il più possibile in piano e ben ancorata al telaio.
- Alla fine è possibile utilizzare una spugna adesiva tra il telaio e la FCU per minimizzare l'impatto delle vibrazioni meccaniche provenienti dai motori.
- L'attuale firmware supporta la configurazione a "X" (la più utilizzata). Se si desidera utilizzare un'altra configurazione sono necessarie alcune modifiche al firmware.

Passo 2: Collegamento dei Motori

16

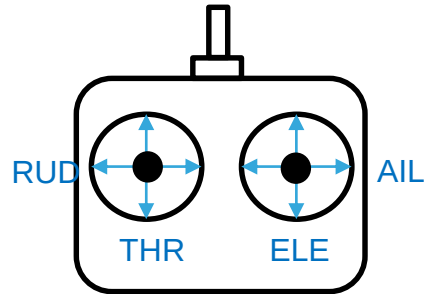
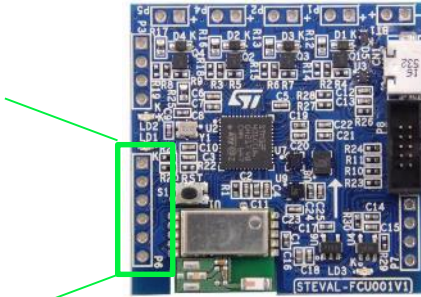


- I motori brushed DC (o i motori brushless collegati all'ESC esterno) devono essere collegati seguendo l'ordine della figura.
- Di solito i motori brushed DC hanno i fili di colore diverso per "+" e "-" e per CW e CCW (fare riferimento alla documentazione del motore).
- **Attenzione:** non montare ancora le eliche !! Prima di montarle è necessario eseguire un controllo completo del sistema per evitare possibili danni o lesioni.
- **Attenzione:** un motore CW DC può ruotare anche in senso antiorario se "+" e "-" sono invertiti, ma le spazzole sono progettate per CW e l'utilizzo del motore in CCW può ridurre significativamente la sua durata.
- **Attenzione:** se i motori sono montati come nella figura ma la direzione di volo della FCU è diversa, quando si cercherà di pilotare il drone questo si capovolgerà immediatamente.

Passo 3: Ricevitore R/C

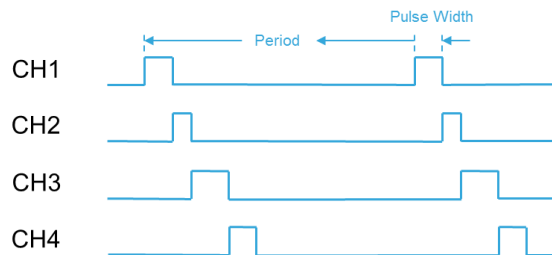
17

PWM input:
Vbat
CH1
CH2
CH3
CH4
GND



- Nel caso in cui si utilizza un Controllo remoto, il modulo RX deve essere collegato all'ingresso PWM (Pulse Width Modulation) della FCU.
- L'utente deve verificare che il trasmettitore sia programmato per avere i 4 canali RC assegnati come da tabella.
- L'attuale FW è compatibile con il ricevitore PPM (Pulse Position Modulation).

La figura si riferisce al telecomando Type2, comunemente usato per quadricotteri. In caso di Type1 i comandi sono invertiti (DX-SX).



Period: ~5..20ms
(50..200Hz)
Pulse Width: 1~2ms

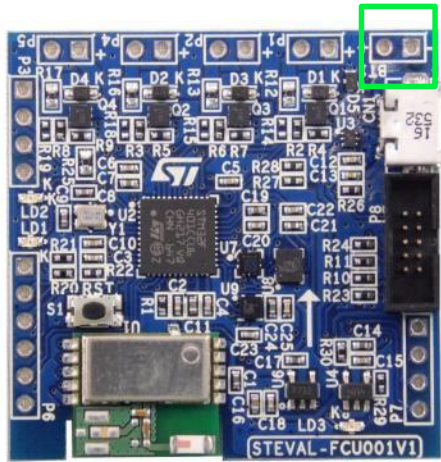
Channel	Control	Function	Position	L / D	M	R / U
CH1	AIL	Rollio - Roll	R: LR	2ms	1.5ms	1ms
CH2	ELE	Beccheggio - Pitch	R: UD	1ms	1.5ms	2ms
CH3	THR	Spinta - Thrust	L: UD	1ms	-	2ms
CH4	RUD	Imbardata - Yaw	L: LR	2ms	1.5ms	1ms

L: Left R: Right U: Up D: Down

Passo 4: Collegamento della Batteria

18

Connettore BT1
Vbat input

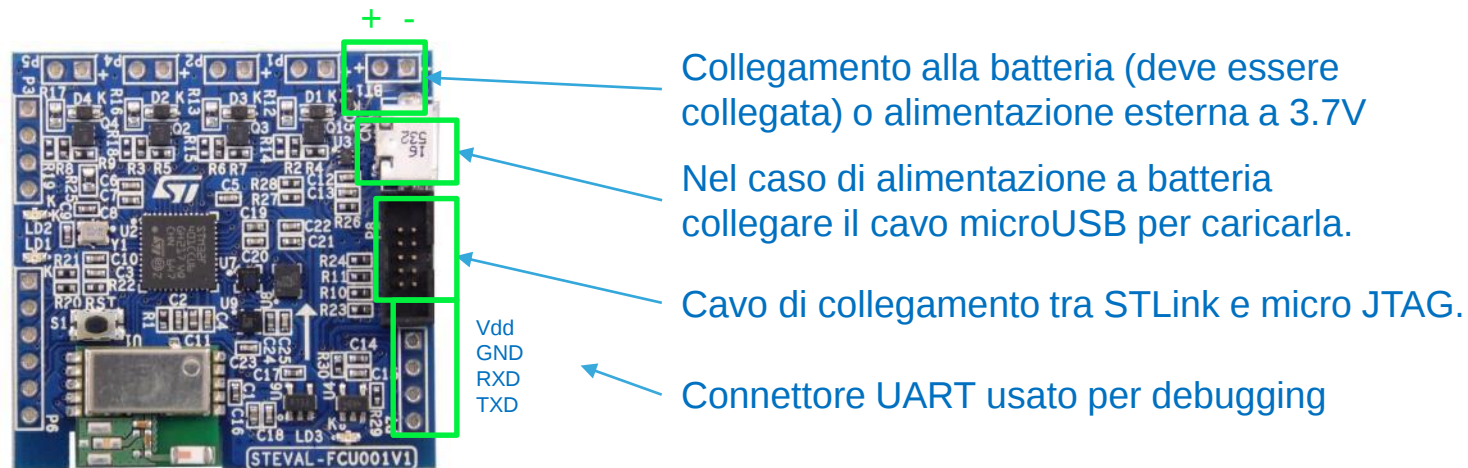


- Una batteria LiPO a 1 cella può essere collegata direttamente al connettore BT1 sulla scheda STEVAL-FCU001V1 e la batteria può essere caricata collegando il cavo USB grazie al caricatore di bordo.
- Nel caso di 2 celle o 3 celle, la batteria deve essere collegata all'ESC e Vbat deve essere collegato alla 5V (tensione di ingresso massima 5,5 V) generata dal BEC dell'ESC.
- **Attenzione:** il diodo di protezione d'inversione di polarità non è montato, si prega di controllare attentamente prima di collegare la batteria!
- **Avvertenza:** la batteria LiPO può essere danneggiata e anche esplodere se cortocircuitate o sovraccaricate. Devono inoltre essere sempre riposte in una borsa quando non utilizzati o durante la ricarica. Leggere le informazioni dettagliate (dal produttore di batterie o su Internet) prima di maneggiare una batteria LiPO.
- **Attenzione:** non appena la batteria si scarica, nel caso in cui l'elica del motore continua a girare ma la velocità non è sufficiente per decollare, scollegare immediatamente la batteria dalla FCU. Se rimane collegata e si scarica completamente, sarà danneggiata irrimediabilmente e non sarà più possibile ricaricarla. (Vedere **Appendice: Caricabatteria integrato per Batterie LiPO**)

Passo 5: Collegamento JTAG e download del FW

19

- Per scaricare e mettere a punto il FW sul MCU STM32F401 è necessario collegare un JTAG con micro connettore (SWD) e fornire una tensione di alimentazione alla scheda tramite una batteria LiPo (3,7 V). Per evitare di scaricare completamente la batteria durante la sessione di debug, si consiglia di collegare sempre il cavo microUSB (collegato al PC o al caricabatterie) per garantire che la batteria sia sempre carica.



Nota: se la batteria, o la tensione di alimentazione, non è collegata al connettore BT1, ma solo al cavo microUSB, la tensione di alimentazione dell'STM32 potrebbe essere instabile e non funzionare correttamente durante la sessione di debug.

Passo 6: Procedure Iniziali di Calibrazione

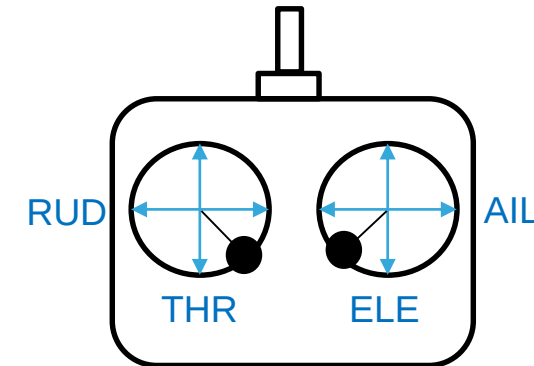
20

- Dopo l'avvio (con la FCU già montata sul telaio), è necessario eseguire una procedura di calibrazione dei sensori.
- Il quadricottero deve essere posizionato su una superficie piana e, dopo aver premuto il pulsante di ripristino, le levette del telecomando devono essere posizionate come nella figura.
- La FCU prenderà come riferimento la superficie piatta e compenserà un possibile spostamento dell'angolo di Eulero mediante l'AHRS (Attitude & Heading Reference System, Sistema di Riferimento per l'Assetto e la Direzione di Volo).

euler_ahrs	<struct>
thx	2.7870795133E-1
thy	7.9544568433E-3
thz	0.0



euler_ahrs	<struct>
thx	4.0701283144E-3
thy	-3.6119404833E-4
thz	0.0

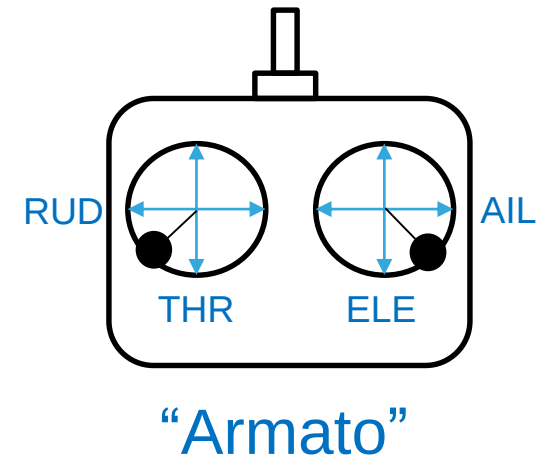


“Calibrazione”

Passo 7: Accensione dei Motori

21

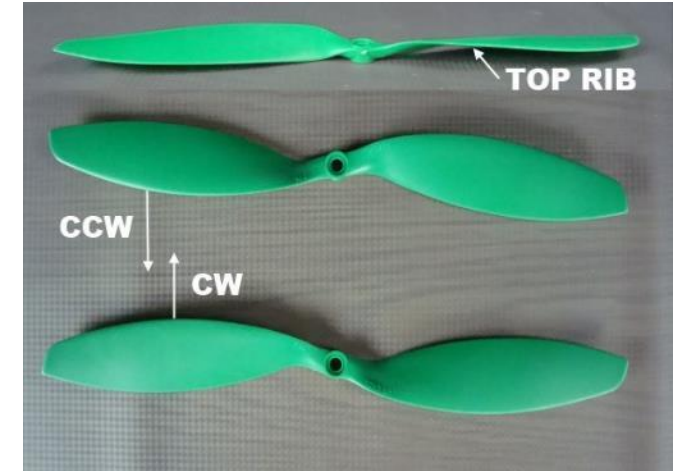
- Prima di montare le eliche e fare un primo test di volo, si consiglia vivamente di fare un controllo senza di esse al fine di verificare che i collegamenti siano corretti e che non ci siano problemi con le connessioni ai motori o al telecomando.
- Per ragioni di sicurezza, il drone è inizialmente «**Disarmato**»: anche se viene stabilita la connessione con il telecomando e la «manopola del gas» è in fondo scala, i motori non vengono azionati intenzionalmente (il LED rosso su STEVAL-FCU001V1 lampeggia).
- Per consentire il volo è necessario completare una cosiddetta procedura di inizializzazione (in gergo «**Armare**»): nell'attuale implementazione FW la procedura è completata dopo aver orientato le manopole del telecomando come mostrato per almeno 2 secondi (il LED rosso smetterà di lampeggiare e rimarrà acceso).



Passo 8: Montaggio delle Eliche

22

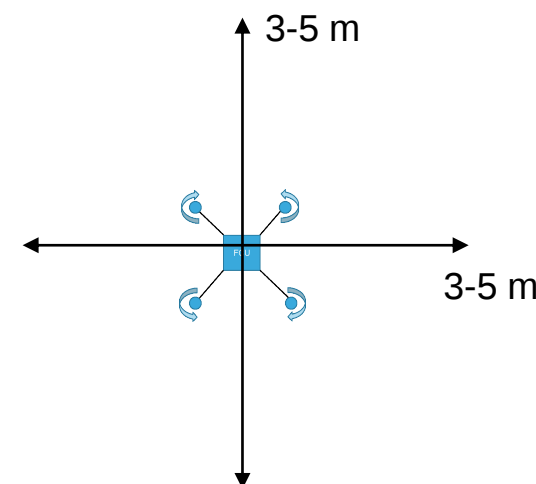
- Una particolare attenzione deve essere rivolta al montaggio corretto dell'elica: l'elica è progettata (seguendo il principio di Bernoulli) per dare una spinta positiva perpendicolare ai suoi assi quando ruota con la «nervatura» superiore che avanza nel senso di rotazione.
- È possibile riconoscere le eliche per l'uso in senso orario o antiorario osservando la direzione della «nervatura» superiore (anche se di solito le eliche antiorarie sono contrassegnate con una "R", invertita).
- **Attenzione:** prima di montare le eliche, scollegare la batteria per sicurezza.
- **Avvertenza:** se si commette un errore e si montano eliche «orarie» su un motore a rotazione in senso antiorario, e viceversa, semplicemente il quadricottero non volerà anche se i motori girano a piena potenza.



Passo 9: Primo Volo di Prova

23

- Per il primo volo di prova, assicurarsi di aver preso tutte le precauzioni di sicurezza e che nessuna persona si trovi nelle vicinanze dell'area di volo. Dopo aver seguito la procedura di inizializzazione sarà possibile decollare e pilotare il drone.
- All'inizio fare volare il drone ad un paio di metri di altezza e assicurati che sia ben stabilizzato.
- Quindi compiere un semplice esercizio di volo come fare un disegno a croce a 2-3 m di altezza.
- **Risoluzione di problemi:** se il drone sembra instabile e inizia a oscillare, provare a ridurre inizialmente il guadagno proporzionale PID.



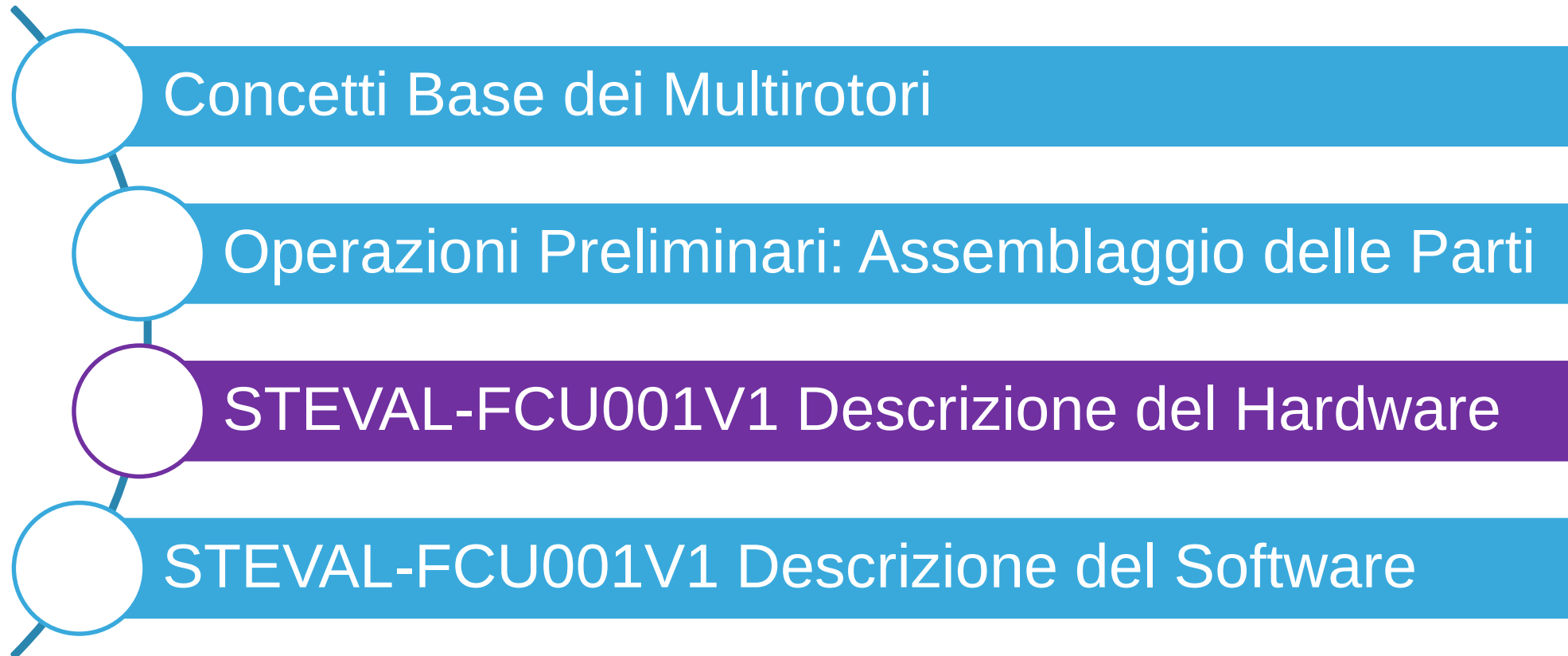
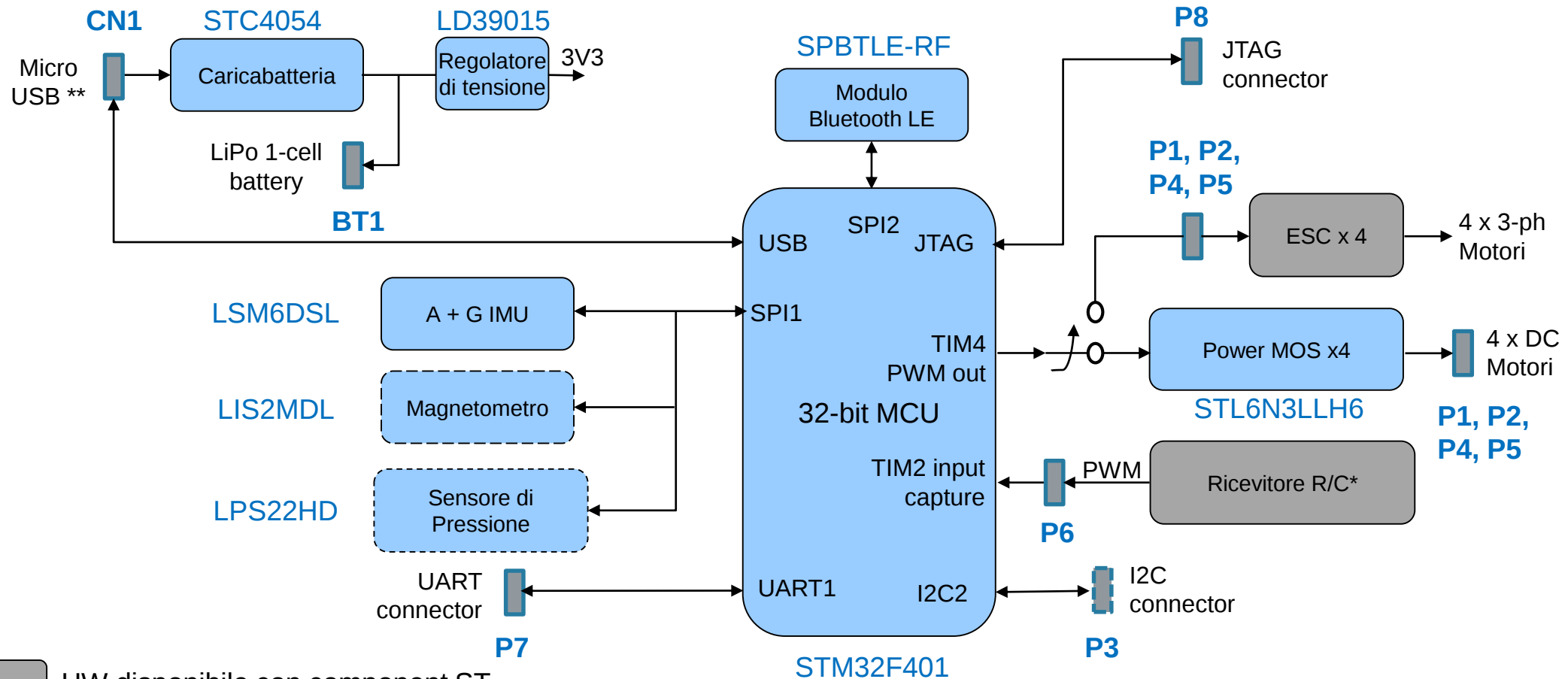


Diagramma a Blocchi STEAVLFCU001V1

25



HW disponibile con component ST.



Sensori presenti sulla board ma non utilizzati nella versione corrente del FW.

(*) Modulo RX da accoppiare al telecomando.

(**) Connettore Micro-USB utilizzato solo per la ricarica della batterie.

La linea dati della porta USB è connessa al microcontroller, ma il driver USB non è incluso nel FW.

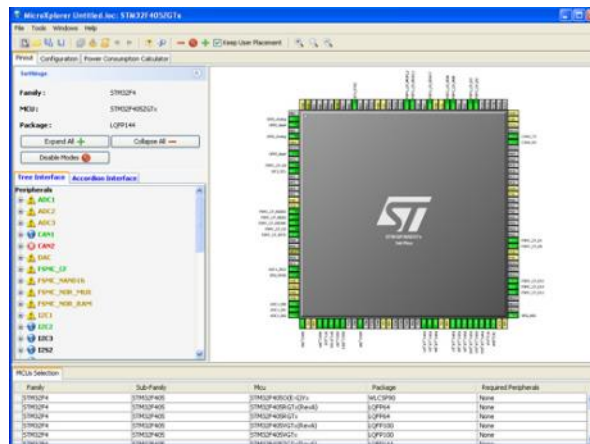
Collegamenti con STM32F4

26



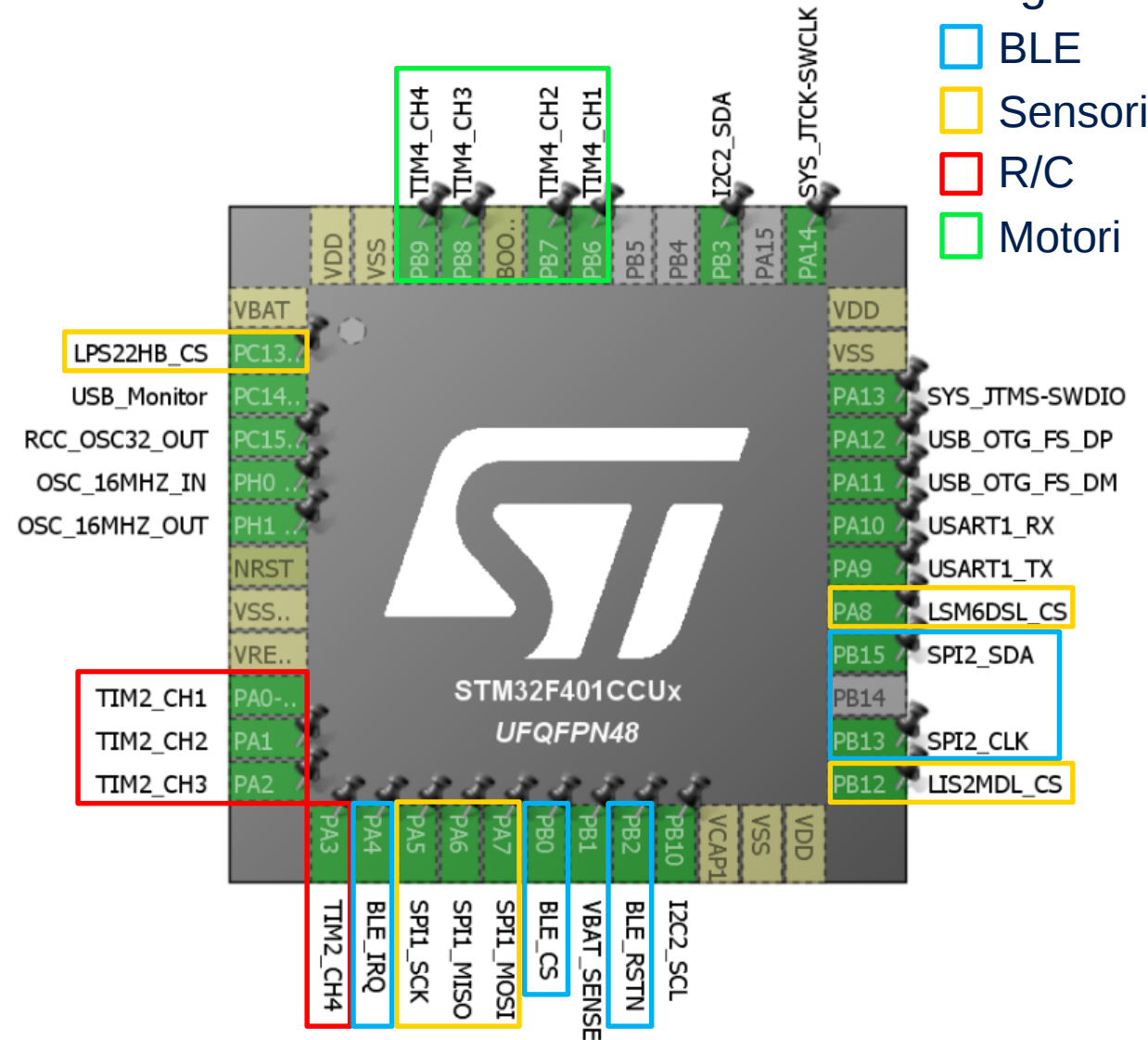
- STM32CubeMX:

- È uno strumento grafico di configurazione software che consente la generazione di codice C mediante procedure guidate.
- Incorpora inoltre pacchetti MCU STM32Cube completi, forniti per serie di microcontroller STM32 (come STM32CubeF4 per serie STM32F4)



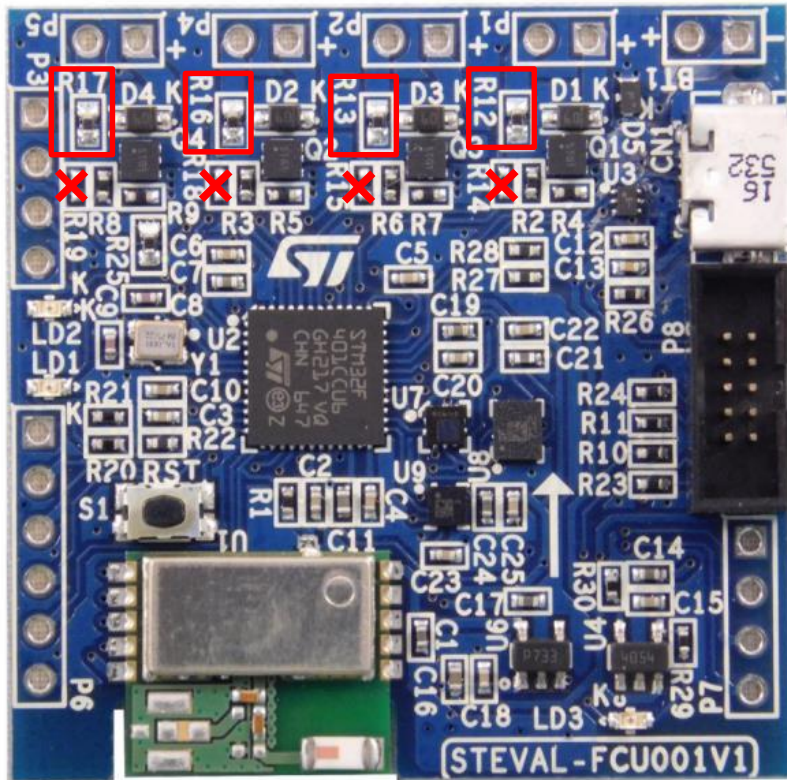
Collegamenti

-  BLE
-  Sensori
-  R/C
-  Motori

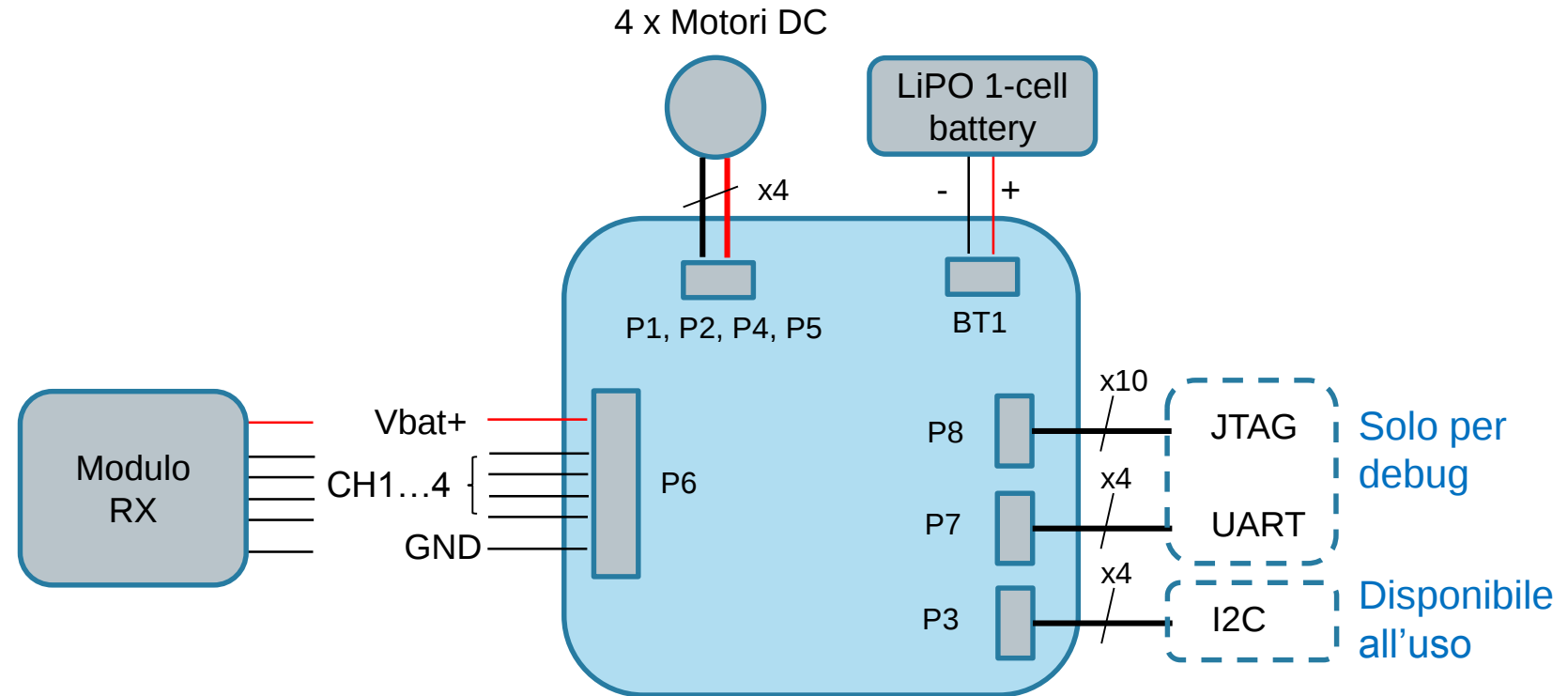


STEVAL-FCU001V1 (Motori DC brushed)

27



Configurazione per Motori a spazzole
R12, R13, R16, R17 SALDATE
R14, R15, R18, R19 aperte

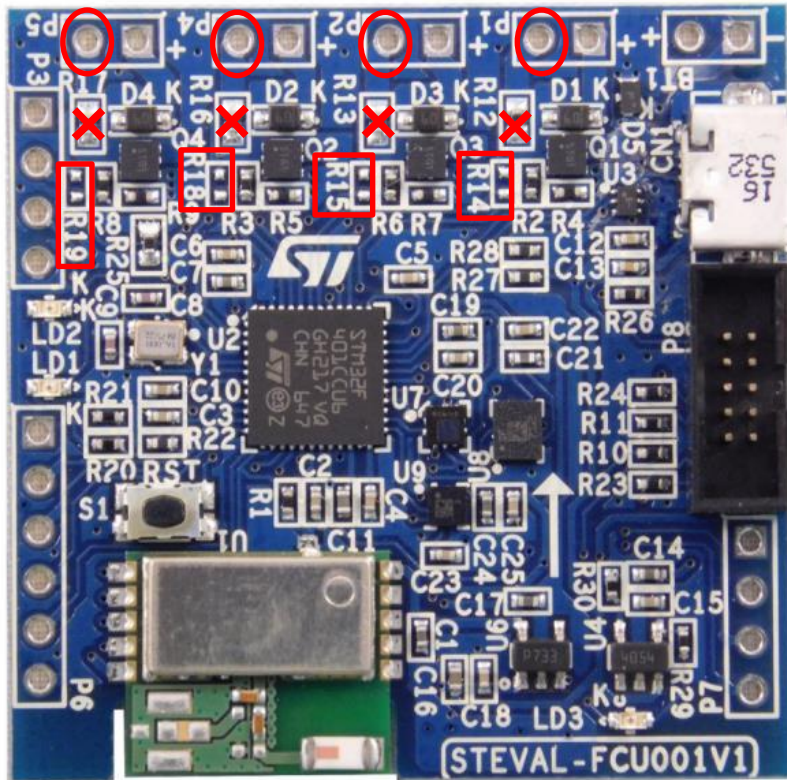


STEVAL-FCU001V1

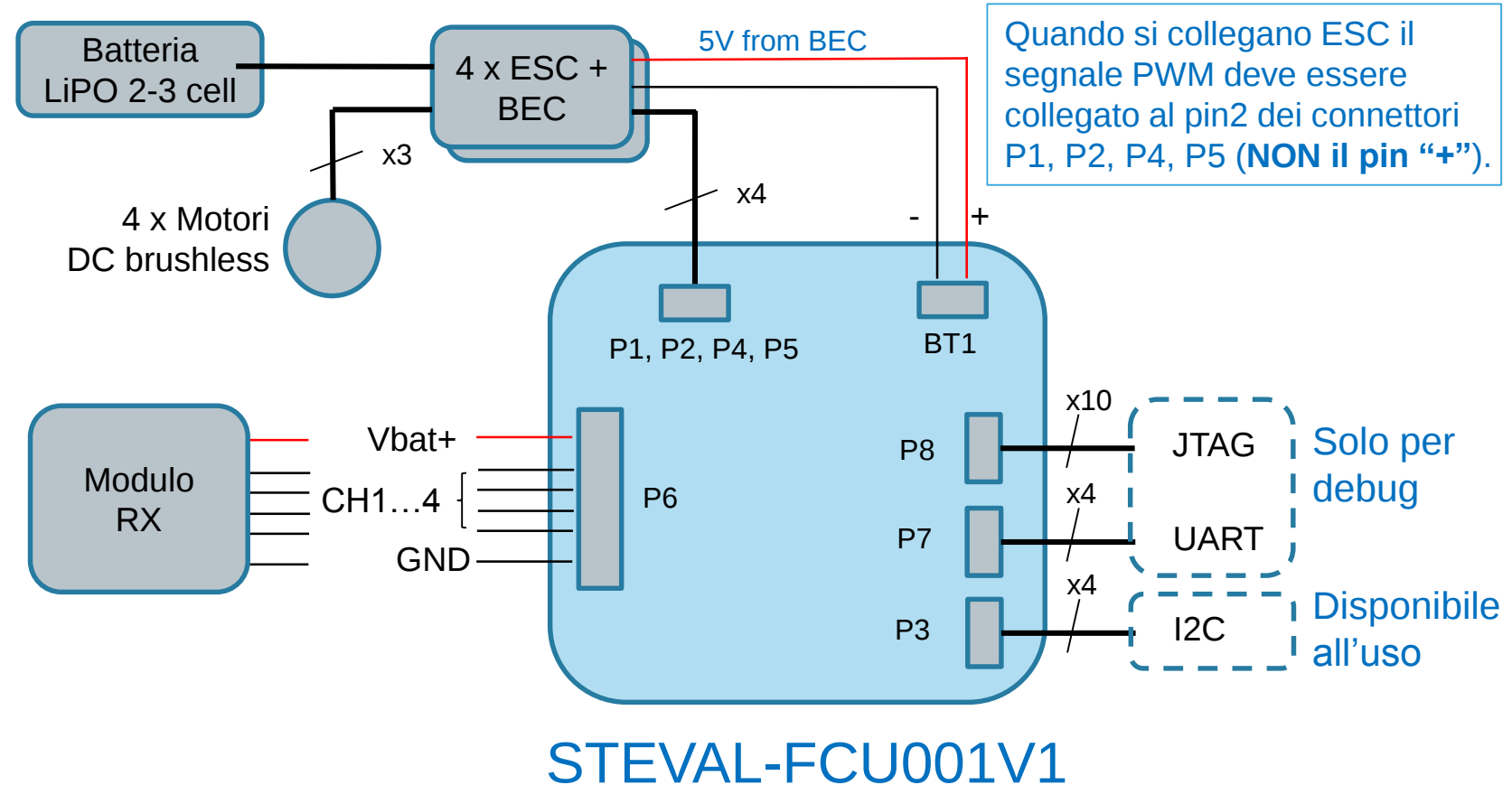
CONFIGURAZIONE DI DEFAULT

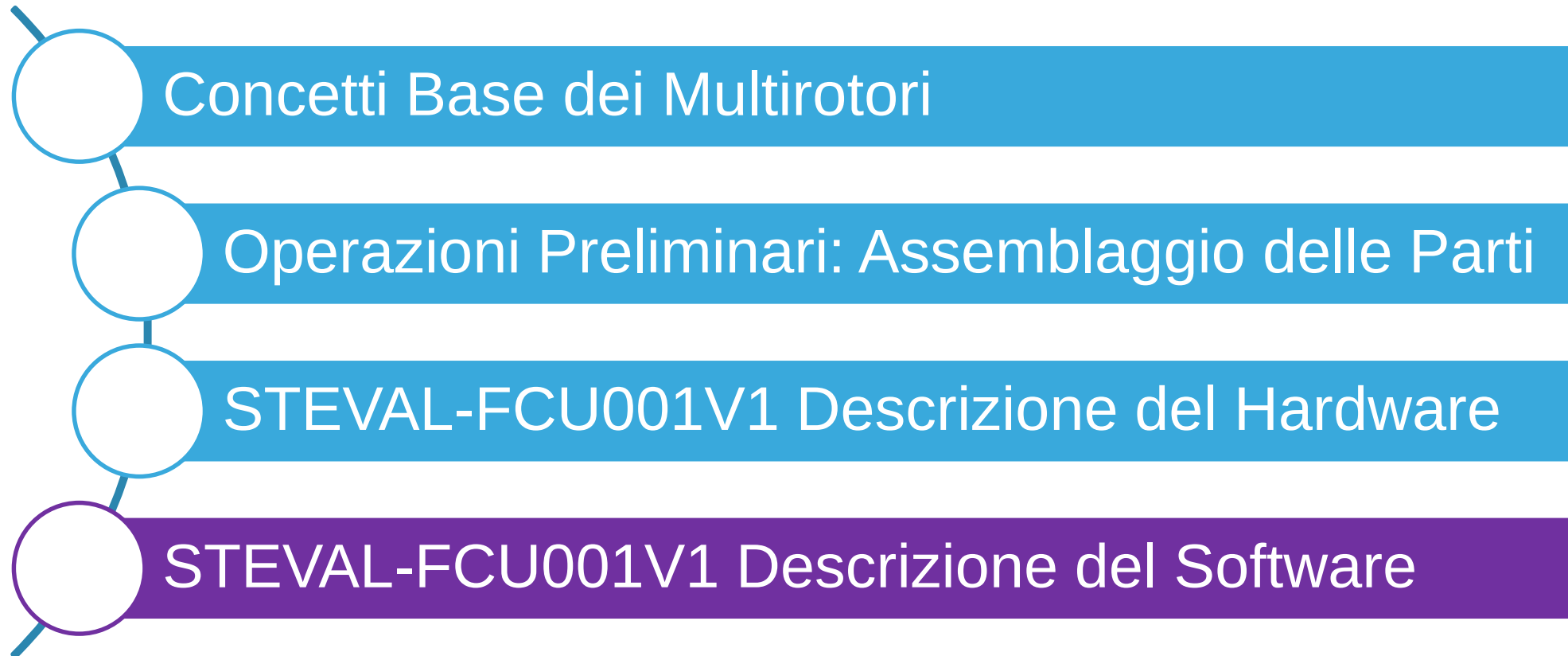
STEVAL-FCU001V1(DC brushless + ESC)

28



Configurazione per Motori Brushless
R12, R13, R16, R17 aperte
R14, R15, R18, R19 SALDATE





STEVAL-FCU001V1 FW download

30

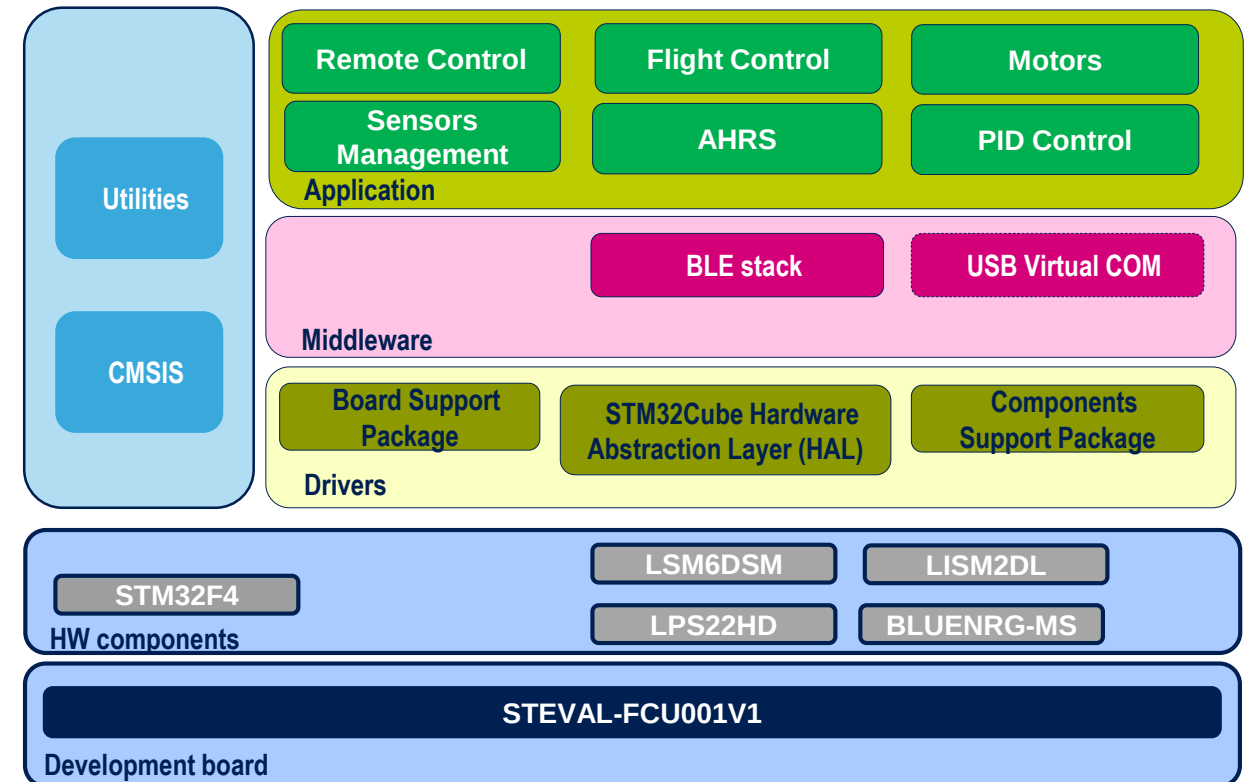
Il codice sorgente del FW della STEVAL-FCU001V1 può essere scaricato al seguente link (o codice QR):

https://github.com/STMicroelectronics-CentralLabs/ST_Drone_FCU_F401

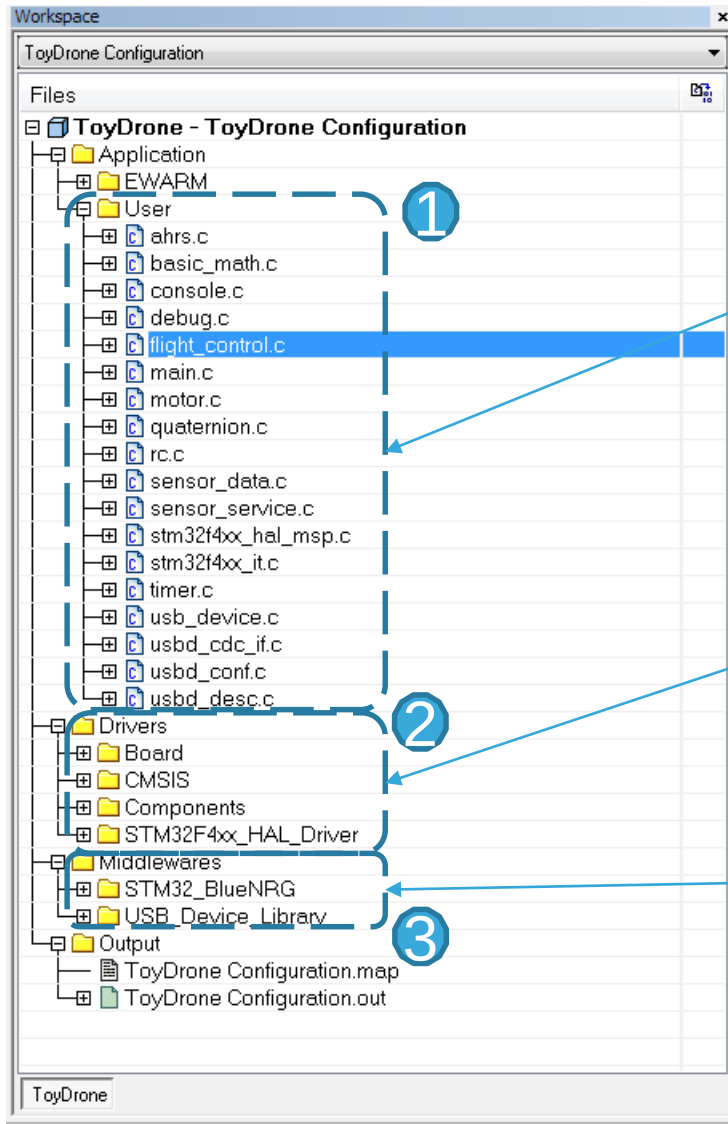


Avvertenze: Il sorgente FW è rilasciato come codice di riferimento allo scopo di valutazione della board, deve essere utilizzato e/o modificato da persone esperte nel campo. Il FW non è stato concepito e testato per uso commerciale o per la produzione in serie di dispositivi.

- Il firmware è composto da diversi livelli:
 - Application:** Contiene tutte le funzioni sviluppate dall'utente per il controllo del volo.
 - Middleware:** sono le librerie che gestiscono le porte di comunicazione.
 - Drivers:** il livello più vicino al HW che contiene le funzioni che gestiscono direttamente i sensori e il microcontrollore.
- Le *Utilities* e *CMSIS* sono funzioni direttamente legate alla architettura STM32 e al relativo Cortex-M4.



 Modulo non ancora implementato nella versione corrente del firmware.



Cartella User:

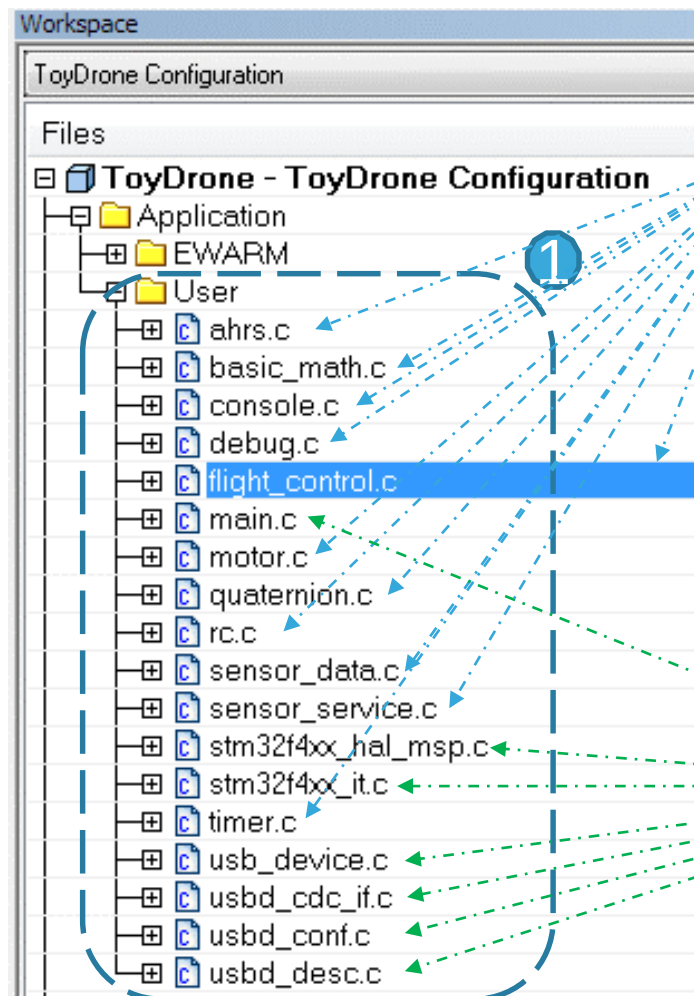
E' la cartella principale del progetto. La maggior parte dei files dedicati al volo e alle comunicazioni sono contenuti in questa cartella. La descrizione del FW in questa presentazione sarà concentrata su questa parte del FW.

Cartella Drivers:

Contiene i drivers dei sensori, della board, e delle HAL (Hardware Abstraction Layer).

Cartella Middlewares:

Contiene le librerie della porta USB e del Bluetooth Low Energy.

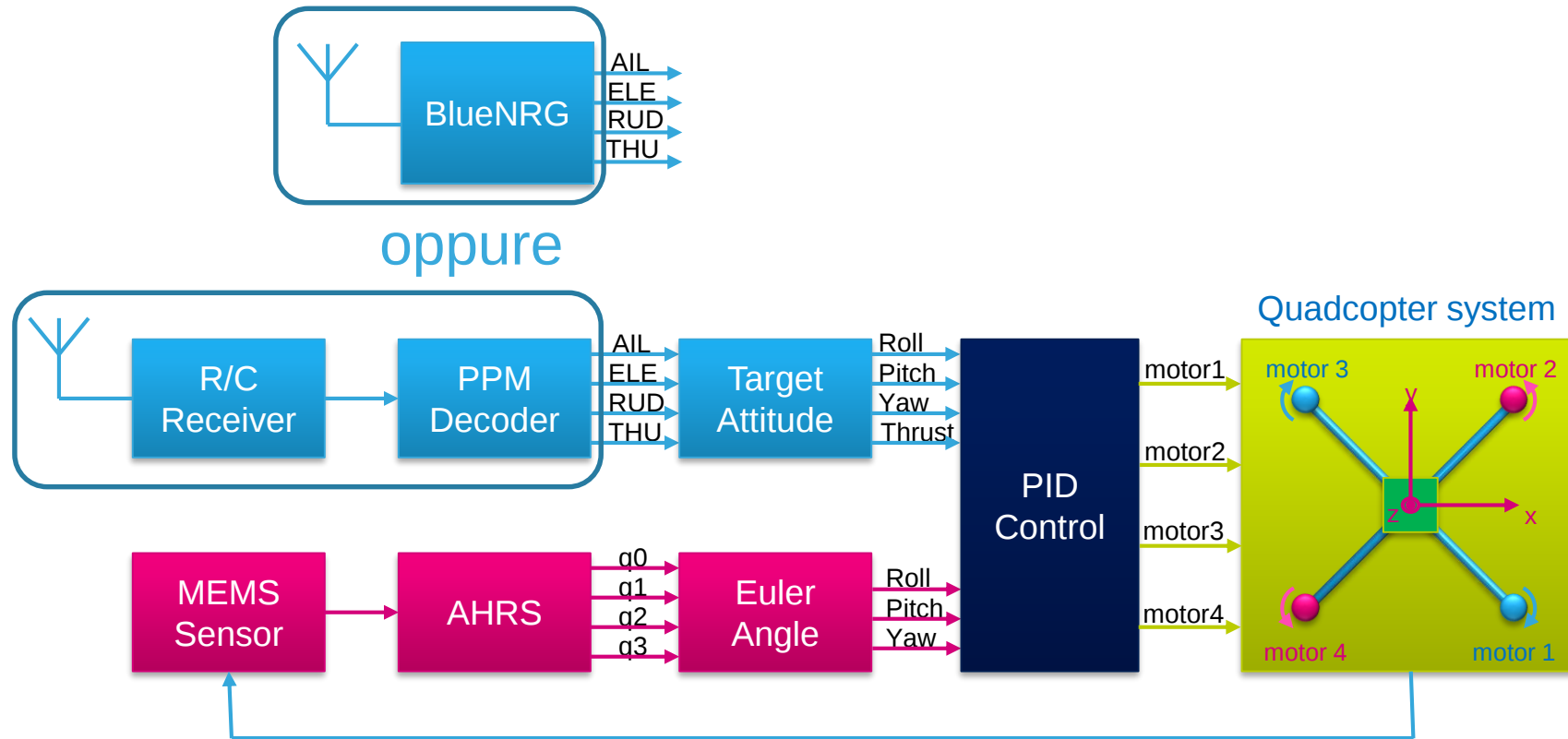


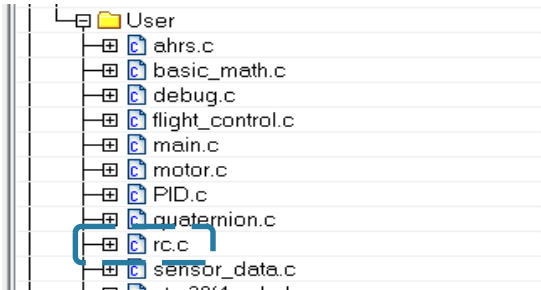
File Specifici sviluppati per l'applicazione Drone

File Standard STM32CUBE

Diagramma a Blocchi

34

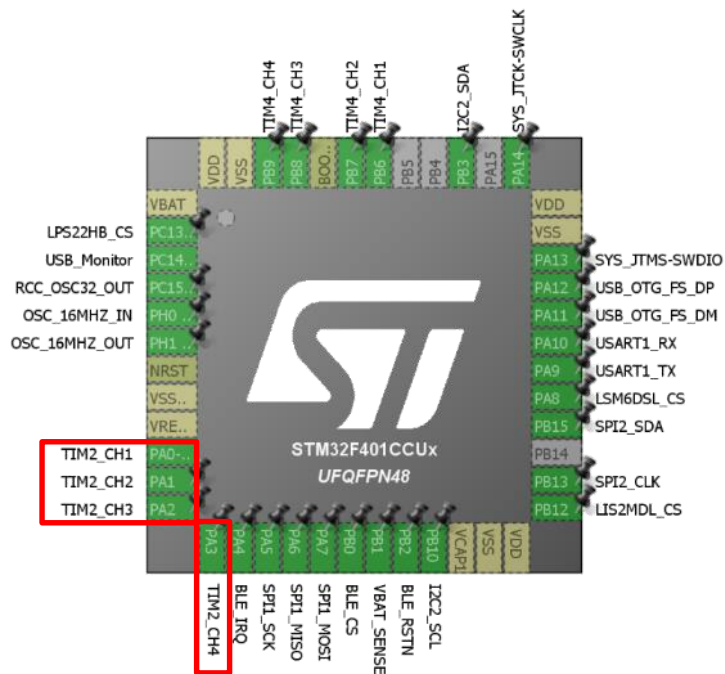




- `volatile int rc_timeout;` >> Contatore di Timeout
- `char rc_connection_flag;` >> Status della connessione

Canali R/C e Calibrazione R/C

36

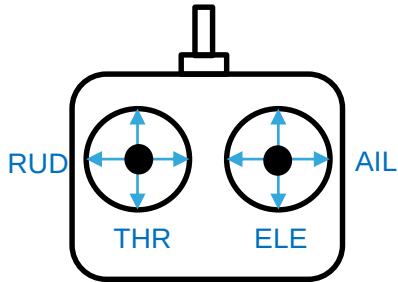


- I canali R/C sono mappati nel file «config_drone.h», dove è anche possibile definire l'intervallo di timeout (se nessun segnale proviene dal telecomando si spengono i Motori):

```
#define RC_TIMEOUT_VALUE      30
// Define the GPIO port for R/C input capture channels
#define RC_CHANNEL1_PORT      GPIOA
#define RC_CHANNEL1_PIN       GPIO_PIN_0
...
```

- È anche possibile calibrare il telecomando regolando con precisione alcuni parametri (posizione centrale, fondo scala, soglia di inizializzazione) in «rc.h»:

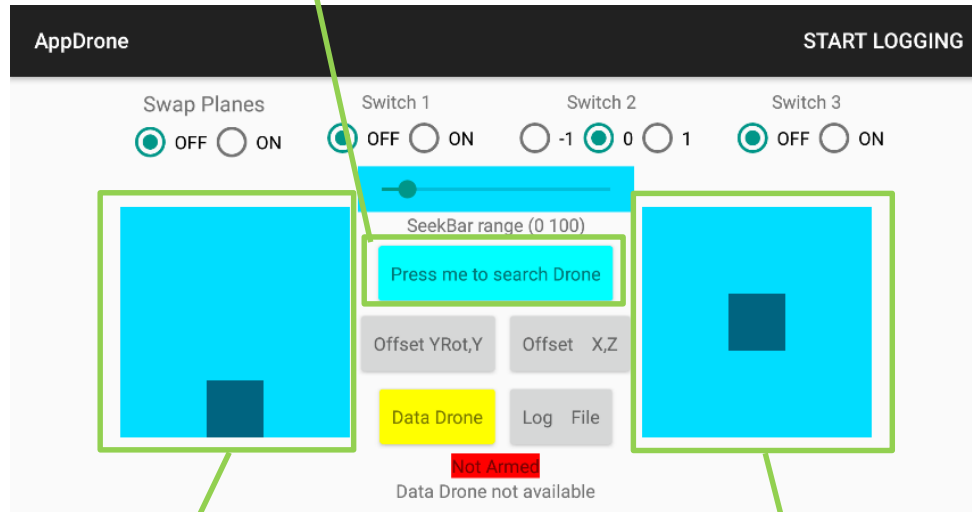
```
// Definition for AIL(Roll) Channel
#define AIL_LEFT      7661
#define AIL_MIDDLE    6044
#define AIL_RIGHT     4434
...
#define RC_FULLSCALE      1500
#define RC_CAL_THRESHOLD  1200
```



Bluetooth ST AppDrone

37

Tasto connessione al Drone



Leva Sinistra (THR, RUD)

Leva Destra (ELE,AIL)

- Nel caso di collegamento con interfaccia BLE valgono tutte le considerazioni fino ad ora fatte per il modulo R/C.

- L'unica differenza sta nell'assenza del modulo PPM Decoder, in quanto il modulo BLE trasmette direttamente i valori delle variabili

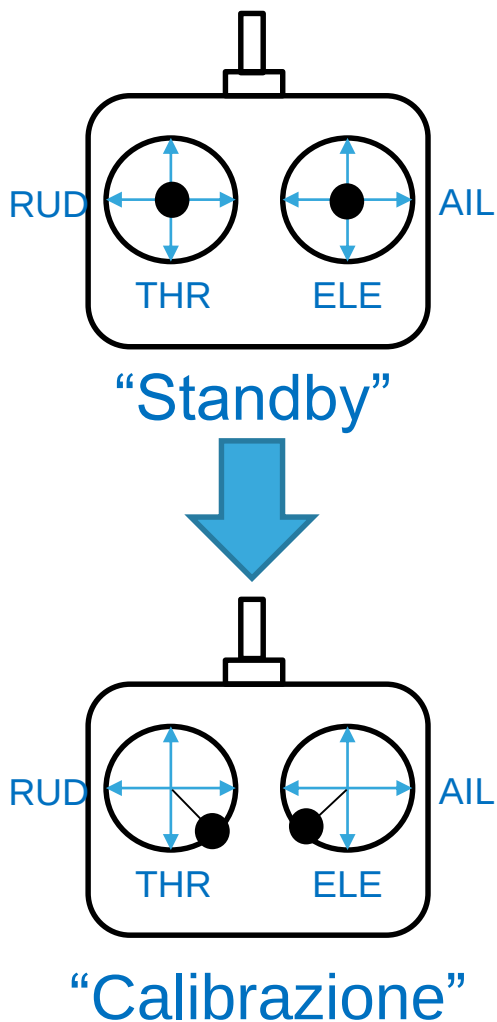
```
int16_t gAIL, gELE, gTHR, gRUD.
```

- Passare dalla versione R/C ad app BLE si attua semplicemente cambiando una `define` nel file `<<rc.h>>`:

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

Procedura di Calibrazione dei Sensori

38

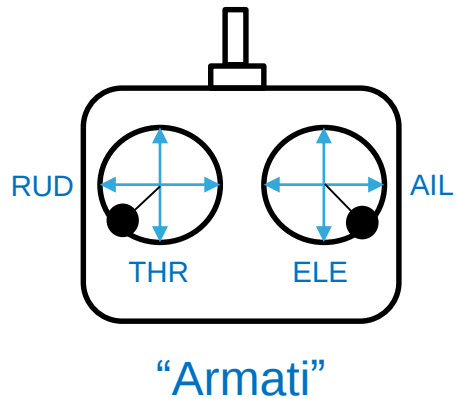
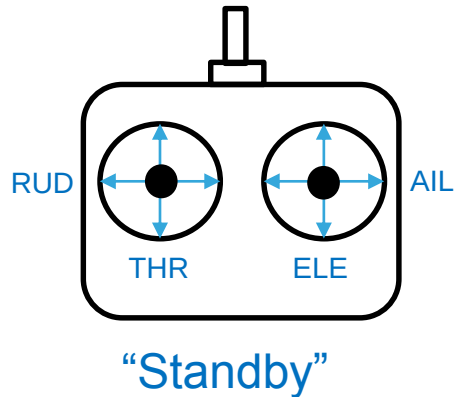


- Se la batteria viene collegata alla FCU (o il pulsante di reset viene premuto) quando il quadricottero non si trova su una superficie piana (o la FCU non è montata in piano sul telaio), l'angolo di Eulero avrà un offset. Per eliminarlo, dopo l'avvio, è possibile eseguire una semplice procedura di calibrazione dal telecomando.
- La procedura di calibrazione può essere personalizzata nella funzione `update_rc_data` nel file «`rc.c`».

```
if ( (gTHR == 0) && (gELE < -RC_CAL_THRESHOLD) && (gAIL > RC_CAL_THRESHOLD) &&
    (gRUD < -RC_CAL_THRESHOLD) )
{
    rc_cal_flag = 1;
}
```

Procedura di Inizializzazione dei Motori

39

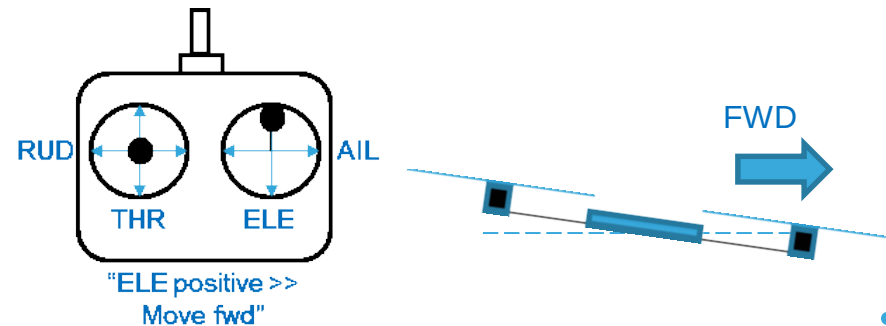
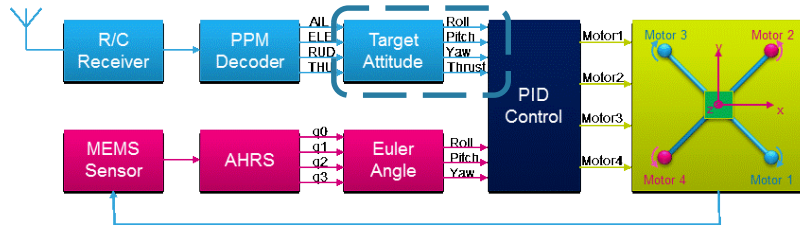


- Come abbiamo già accennato, per motivi di sicurezza (per evitare danni o lesioni quando la batteria del quadricottero è inserita e il telecomando è acceso), viene sempre utilizzata una procedura di inizializzazione: i motori vengono spenti finché non viene applicata una determinata sequenza sul telecomando.
- La procedura di «Arming» può essere personalizzata sempre nella funzione `update_rc_data` nel file «`rc.c`» :

```
if ( (gTHR == 0) && (gELE < -RC_CAL_THRESHOLD) && (gAIL < -RC_CAL_THRESHOLD) && (gRUD > RC_CAL_THRESHOLD))
{
    if (rc_enable_motor==0) // if not armed -> arm
    {
        rc_enable_rotor = 1;
        fly_ready = 1;
    }
    else // if armed -> disarm
    {
        rc_enable_motor = 0;
        fly_ready = 0;
    }
}
```

Conversione degli Angoli di Eulero

40



- La conversione da angoli di Eulero nel sistema di riferimento del telecomando ad angoli di Eulero in AHRS (necessario per impostare il target del loop di controllo PID) è implementata nel file «rc.c» nella funzione:

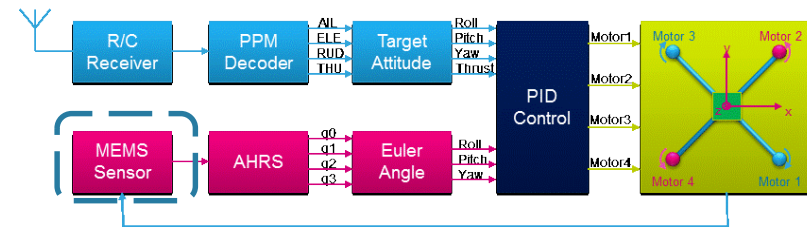
```
void GetTargetEulerAngle(EulerAngleTypeDef *euler_rc,  
                        EulerAngleTypeDef *euler_ahrs)
```

- I dati dal telecomando (cioè gELE) sono normalizzati rispetto al valore di fondo scala e convertiti in angolo:

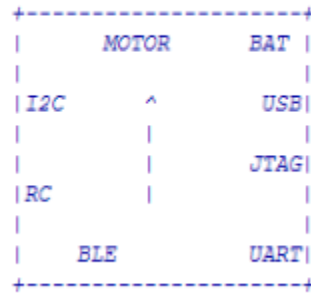
```
const float max_pitch_rad = PI*PITCH_MAX_DEG/180.0;
```

```
t1 = gELE;  
if (t1 > RC_FULLSCALE)  
    t1 = RC_FULLSCALE;  
else if (t1 < -RC_FULLSCALE)  
    t1 = -RC_FULLSCALE;  
euler_rc->thx = -t1 * max_pitch_rad / RC_FULLSCALE;
```

- Il cambiamento di segno (-t1) è legato al fatto che aumentando ELE per muoversi avanti, corrisponde ad un angolo target negativo per il pitch.



STEVAl-FCU001V1



^ (ACC X)
|
(ACC Y) <-- 0(z)

- (1) Drone FWD direction V (ACC -X)
- (2) Drone FWD direction > (ACC -Y)
- (3) Drone FWD direction ^ (ACC X)
- (4) Drone FWD direction < (ACC Y)

Translation of coordinates for AHRS
(1) Drone FWD direction > (ACC -X)



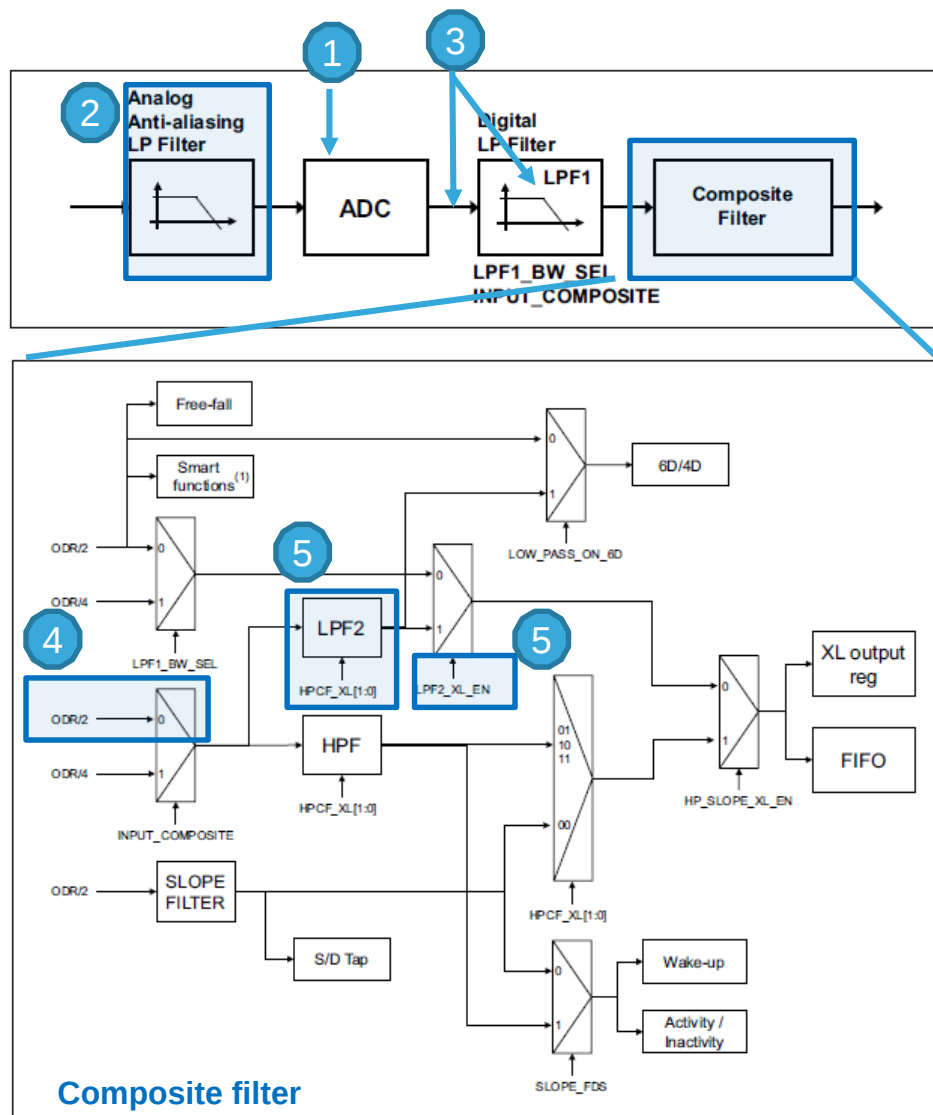
(x) <-- 0(z)
|
V(y)

- I principali sensori utilizzati per la stabilizzazione del volo sono l'accelerometro e il giroscopio (LSM6DSL). I driver dei sensori sono implementati nel file «Components\LSM6DSL_ACC_GYRO_driver.c»
- Nel progetto FW sono presenti anche i driver per i sensori di magnetometro e pressione (LIS2MDL e LPS22HB), ma non sono utilizzati nell'attuale implementazione FW.
- L'inizializzazione dei sensori viene eseguita nel file «main.c»:

```
/* Initialize and Enable the available sensors on SPI*/  
initializeAllSensors();  
enableAllSensors();
```
- I dati RAW provenienti dai sensori Acc + Gyro a 6 assi vengono quindi convertiti in unità [mg] e [mdps] e tradotti nel sistema di coordinate utilizzato (orientamento della scheda FCU rispetto alla configurazione dei motori) in «sensor_data.c».

Setup dell'Accelerometro

42



- Tutto l'algoritmo di controllo del volo si basa sui dati generati da accelerometro e giroscopio.
- È importante assicurarsi che i sensori siano correttamente messi a punto.
- In particolare è importante che i dati RAW non siano influenzati dal rumore meccanico generato dalla vibrazione dei Motori e propagato attraverso il telaio.

1. Selezione del fondo scala (FS)
2. Larghezza di banda del filtro analogico
3. Velocità di trasmissione dei dati (ODR)
4. Selezione del input del filtro composito
5. Selezione del filtro passa basso LPF2

Setup dell'Accelerometro

43

- Impostazione dei parametri principali dell'accelerometro:
 - Selezione del fondo scala:
 - *default: $\pm 2 g$ -> impostare $\pm 4 g$*
 - Larghezza di banda del filtro analogico:
 - *Default: 1.5KHz -> usare valore predefinito*
 - Velocità di trasmissione dei dati (ODR) e selezione power mode:
 - *default: Power down mode -> impostare a 1.6kHz*
 - Selezione del input del filtro composito :
 - *default: ODR/2 -> usare valore predefinito*
 - Filtro passa basso LPF2 dell'accelerometro:
 - *impostare il Filtro passa basso abilitato a ODR/50 o ODR/100 in base alla vibrazione meccanica rilevata*

- Nel codice abbiamo la seguente corrispondenza con quanto descritto:

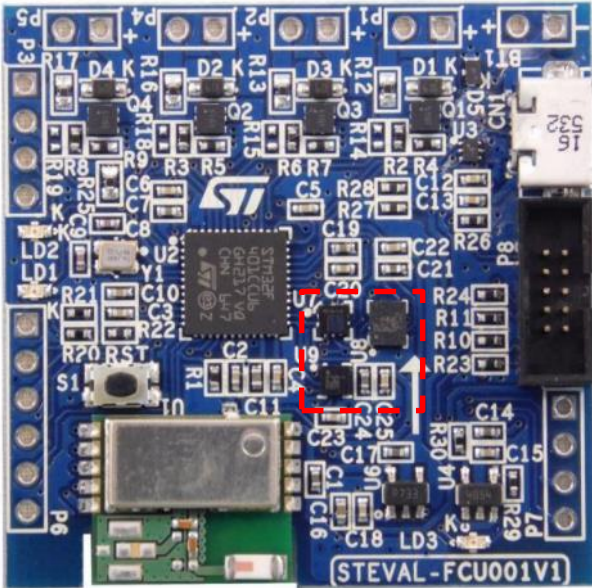
```
BSP_ACCELERO_Set_ODR_Value(LSM6DSL_X_0_handle, 1660.0);           /* ODR 1.6kHz */
BSP_ACCELERO_Set_FS(LSM6DSL_X_0_handle, FS_MID);                 /* FS 4g */
// ODR/2 low pass filtered sent to composite filter
LSM6DSL_ACC_GYRO_W_InComposit(LSM6DSL_X_0_handle, LSM6DSL_ACC_GYRO_IN_ODR_DIV_2);
// Enable LPF2 filter in composite filter block
LSM6DSL_ACC_GYRO_W_LowPassFiltSel_XL(LSM6DSL_X_0_handle, LSM6DSL_ACC_GYRO_LPF2_XL_ENABLE);
// Low pass filter @ ODR/100
LSM6DSL_ACC_GYRO_W_HPCF_XL(LSM6DSL_X_0_handle, LSM6DSL_ACC_GYRO_HPCF_XL_DIV100); // Set
LSB to 0 >> Analog filter 1500Hz
uint8_t tmp_6axis_reg_value;
BSP_ACCELERO_Read_Reg(LSM6DSL_X_0_handle, 0x10, &tmp_6axis_reg_value);
tmp_6axis_reg_value = tmp_6axis_reg_value & 0xFE;
BSP_ACCELERO_Write_Reg(LSM6DSL_X_0_handle, 0x10, tmp_6axis_reg_value);
```

- Parametri fondamentali del Giroscopio:
 - Selezione del fondo scala (FS):
 - *Default: 245 dps >> settare a 2000dps*
 - Velocità di trasmissione dei dati (ODR):
 - *Default: power down >> settare a 104Hz*
 - Filtro passa basso LPF1 FTYPE :
 - *Settare a "Narrow"*
- Nel codice abbiamo la seguente corrispondenza

```
BSP_GYRO_Set_FS(LSM6DSL_G_0_handle, FS_HIGH); /* Set FS to 2000dps */  
BSP_GYRO_Set_ODR(LSM6DSL_G_0_handle, ODR_HIGH); /* Set ODR to 104Hz */  
LSM6DSL_ACC_GYRO_W_LP_BW_G(LSM6DSL_G_0_handle, LSM6DSL_ACC_GYRO_LP_G_NARROW); /* LPF1 FTYPE set to 10b */
```

Uso del sensore di Pressione e della Bussola

46



- Nel caso in cui sia necessario utilizzare il sensore di pressione (per la stima dell'altitudine) e il magnetometro (per la bussola), è necessario abilitare i sensori nel file «config_drone.h».

```
// Use magnetic sensor or not 1 = use
#define USE_MAG_SENSOR      0

// Use pressure sensor or not 1 = use
#define USE_PRESSURE_SENSOR 0
```

I dati dei sensori vengono aggiornati con una frequenza di 800 Hz (utilizzando l'interrupt Timer9) in «sensor_data.c» :

```
ReadSensorRawData (&acc, &gyro, &mag, &pre);
```

- Quindi, se è necessario, ad esempio a scopo di debug, si possono visualizzare i dati sulla porta UART mediante una PRINTF :

```
PRINTF("Pressure [atm] = %f\n\n",pre);

PRINTF("Magnetometer X = %d\tY = %d\tZ = %d\n\n", mag.AXIS_X, mag.AXIS_Y, mag.AXIS_Z);
```

- I dati di pressione sono espressi in atmosfere, mentre i dati del magnetometro sono in gradi (rispetto al polo nord magnetico).

Calibrazione dei Sensori dopo l'avvio

47

Expression	Value	Location
pre	1.03042163E+3	0x20000B38
p_id	Error (col 1): Un...	
mag	<struct>	0x200009F8
VBAT_Sen...	Error (col 1): Un...	
VBAT	Error (col 1): Un...	
gELE	-1	0x20000B54
gAIL	-422	0x20000B52
gRUD	199	0x20000B58
gTHR	0	0x20000B56
motor_pwm	<struct>	0x2000097C
motor1_p...	0.0	0x2000097C
motor2_p...	0.0	0x20000980
motor3_p...	0.0	0x20000984
motor4_p...	0.0	0x20000988
euler_ahrs	<struct>	0x200009C8
thx	4.57809686E-1	0x200009C8
thy	-1.09683387E-2	0x200009CC
thz	0.0	0x200009D0
ahrs	<struct>	0x2000094C
pid	<struct>	0x20000498

Expression	Value	Location
pre	1.03048217E+3	0x20000B38
p_id	Error (col 1): Un...	
mag	<struct>	0x200009F8
VBAT_Sen...	Error (col 1): Un...	
VBAT	Error (col 1): Un...	
gELE	0	0x20000B54
gAIL	-422	0x20000B52
gRUD	199	0x20000B58
gTHR	0	0x20000B56
motor_pwm	<struct>	0x2000097C
motor1_p...	0.0	0x2000097C
motor2_p...	0.0	0x20000980
motor3_p...	0.0	0x20000984
motor4_p...	0.0	0x20000988
euler_ahrs	<struct>	0x200009C8
thx	-7.1783561722E-4	0x200009C8
thy	-6.69909815E-4	0x200009CC
thz	0.0	0x200009D0
ahrs	<struct>	0x2000094C
pid	<struct>	0x20000498

- Dopo aver inserito la batteria è importante posizionare il quadricottero su una superficie piana e premere il pulsante di ripristino. Infatti, subito dopo il reset, se il quadricottero ha una certa inclinazione, ci sarà un offset nelle coordinate di riferimento (*euler_ahrs*).
- Ciò accade perché subito dopo l'avvio è eseguita una calibrazione automatica dei sensori.

SBAGLIATO

Eccessiva inclinazione all'avvio >> $\text{euler_ahrs.thx} = 0.457(\text{rad}) * 360/2\pi = 26.19 (\text{deg})$

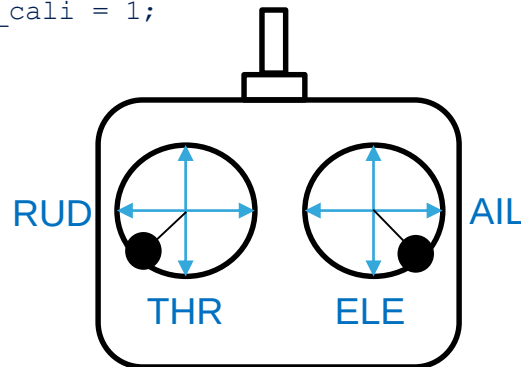
CORRETTO

Inclinazione all'avvio nella norma >> $\text{euler_ahrs.thx} = -0.0371(\text{rad}) * 360/2\pi = -2.12 (\text{deg})$

Calibrazione dei Sensori dopo l'avvio

48

```
if(sensor_init_cali == 0){
    sensor_init_cali_count++;
    if(sensor_init_cali_count > 800)
    {
        // Read sensor data and prepare for specific
        // coordinate system
        ReadSensorRawData(&acc, &gyro, &mag, &pre);
        acc_off_calc.AXIS_X += acc.AXIS_X;
        acc_off_calc.AXIS_Y += acc.AXIS_Y;
        acc_off_calc.AXIS_Z += acc.AXIS_Z;
        gyro_off_calc.AXIS_X += gyro.AXIS_X;
        gyro_off_calc.AXIS_Y += gyro.AXIS_Y;
        gyro_off_calc.AXIS_Z += gyro.AXIS_Z;
        if (sensor_init_cali_count >= 1600)
        {
            acc_offset.AXIS_X = acc_off_calc.AXIS_X * 0.00125;
            acc_offset.AXIS_Y = acc_off_calc.AXIS_Y * 0.00125;
            ...
            sensor_init_cali_count = 0;
            sensor_init_cali = 1;
        }
    }
}
```

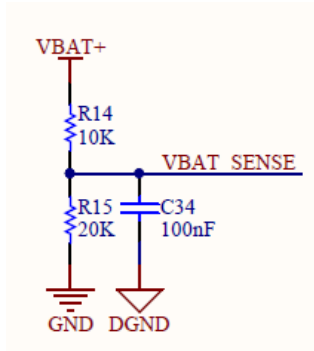


- Guardando nei dettagli, sia il giroscopio che l'accelerometro sono calibrati in base a due condizioni (in `HAL_TIM_PeriodElapsedCallback`) :
 - quando la variabile `sensor_init_cali` è 0, subito dopo l'avvio del sistema (batteria inserita o pulsante di reset premuto).
 - Quando la variabile `rc_cal_flag` è 1, la calibrazione manuale è avviata prima che i Motori siano armati se le leve si trovano in una determinata posizione (controllare la funzione `update_rc_data` nel file «`rc.c`»):

```
if ((gTHR == 0) && (gELE < -RC_CAL_THRESHOLD) &&
    (gAIL > RC_CAL_THRESHOLD) && (gRUD < -RC_CAL_THRESHOLD)) {
    rc_cal_flag = 1;
}
```

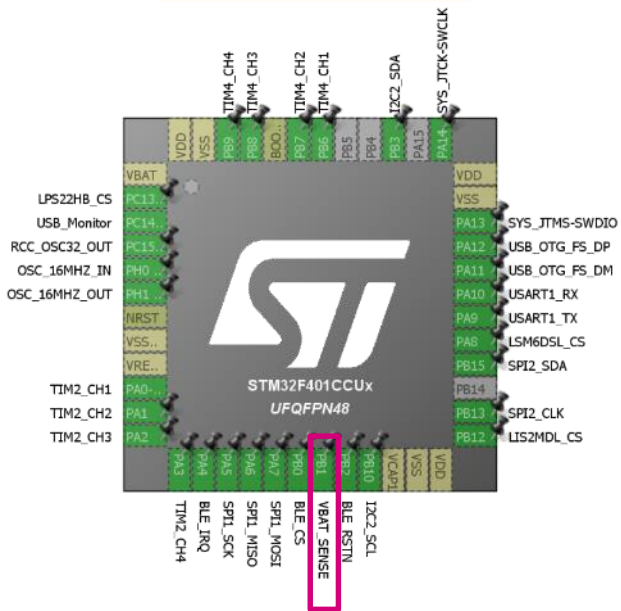
Monitoraggio Batteria

49



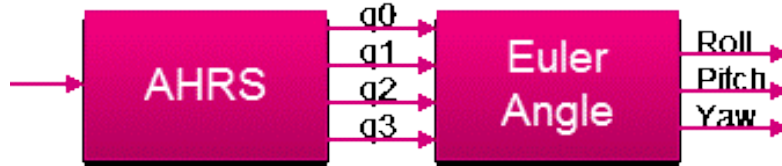
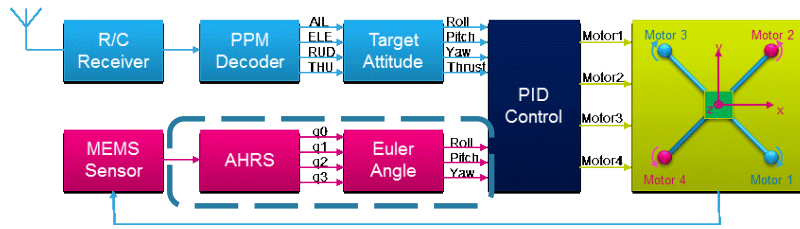
- A volte risulta conveniente poter monitorare il livello della batteria, a scopo di telemetria, o per altri motivi (fermare i motori, avviare un allarme sonoro, ...).
- L'ADC1, canale 9 (PB1), è collegato a tale scopo a VBAT attraverso un partitore di tensione. A scopo di debug, un semplice codice C è stato inserito nel file «main.c» per eseguire una singola lettura ADC (*) e inviare i dati convertiti alla UART:

```
HAL_ADC_Start(&hadc1);  
if (HAL_ADC_PollForConversion(&hadc1, 1000000) == HAL_OK)  
{  
    VBAT_Sense = HAL_ADC_GetValue(&hadc1);  
    VBAT = (((VBAT_Sense*3.3)/4095) * (BAT_RUP+BAT_RDW)) / BAT_RDW;  
    PRINTF("Battery voltage = %fV\r", VBAT);  
}  
HAL_ADC_Stop(&hadc1);
```



Sistema di Riferimento di Assetto e Direzione di volo: AHRS

50



$$\begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(2q_2q_3 + 2q_0q_1, q_3^2 - q_2^2 - q_1^2 + q_0^2) \\ -\text{asin}(2q_1q_3 - 2q_0q_2) \\ \text{atan2}(2q_1q_2 + 2q_0q_3, q_1^2 + q_0^2 - q_3^2 - q_2^2) \end{bmatrix}$$

- I dati RAW da Accelerometro e Giroscopio sono trasferiti all'algoritmo AHRS (funzione presente nel file «ahrs.c»):

```
ahrs_fusion_ag(&acc_ahrs, &gyro_ahrs, &ahrs);
```

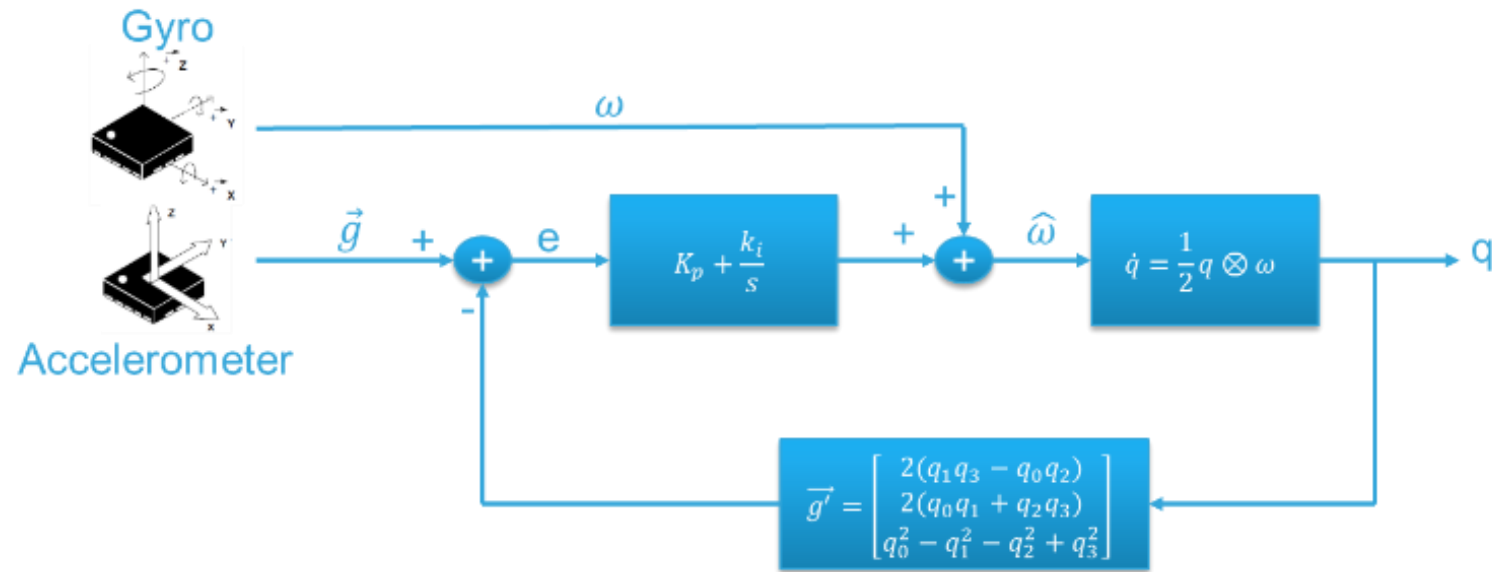
- Per eseguire l'algoritmo di stabilizzazione del drone, i dati del quaternion proveniente dal AHRS devono essere tradotti in angolo di Eulero (funzione presente nel file «quaternion.c»):

```
QuaternionToEuler(&ahrs.q, &euler_ahrs);
```

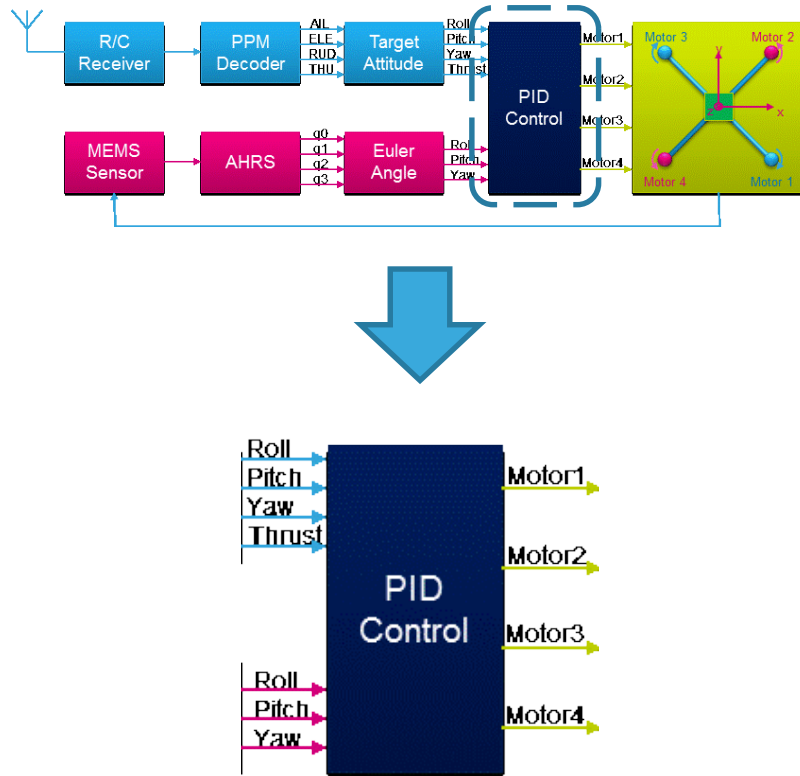
- Il Sistema di Riferimento di Assetto e Direzione di volo (AHRS) è l'algoritmo chiave di un sistema UAV. L'algoritmo scelto per il FW attuale è il "**Filtro Complementare**".

AHRS – Filtro Complementare

51



- L'accelerometro e il giroscopio possono stimare l'assetto in modo indipendente.
 - L'accelerometro è sempre influenzato dal rumore ad alta frequenza e il giroscopio ha un offset a bassa frequenza.
 - I risultati del giroscopio sono precisi nel tempo a breve termine, ma genereranno un offset a lungo termine.
 - L'accelerometro non è preciso a breve termine, a causa del rumore, ma può essere considerato affidabile a lungo termine.
- Il Filtro è basato sul metodo di Mahony.
 - Si utilizza il giroscopio per ottenere l'assetto attuale e si usa l'accelerometro per ricalibrare l'assetto (La misurazione della gravità è usata per stimare la deriva del giroscopio su pitch e roll).



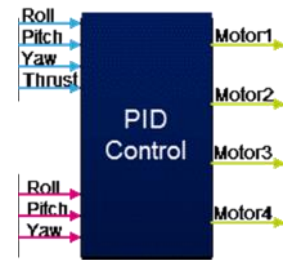
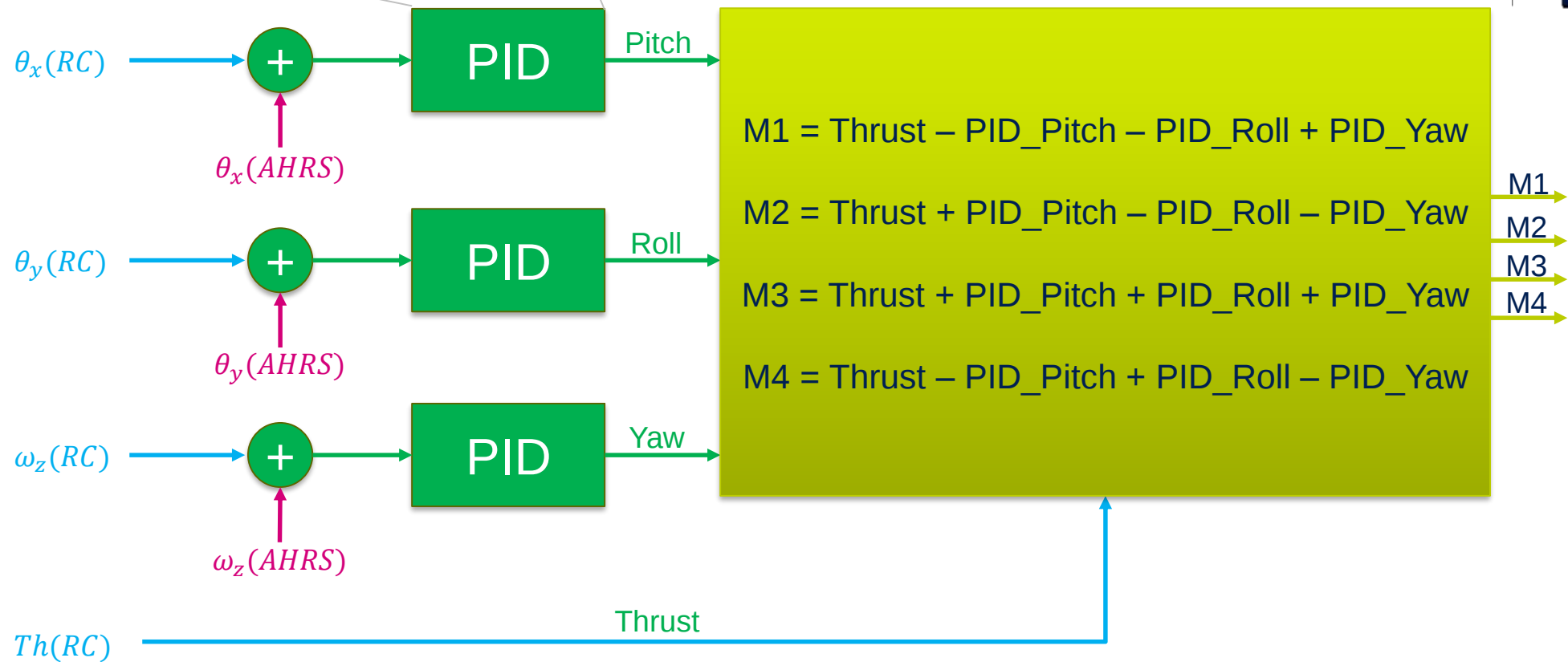
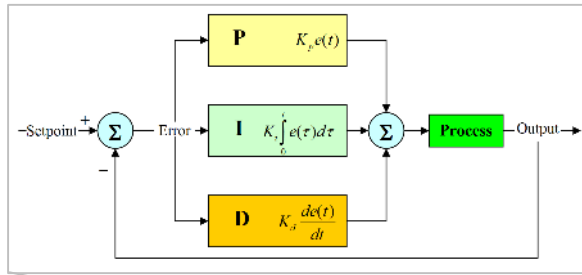
- L'algoritmo di stabilizzazione viene eseguito mediante un controllo PID.
- L'obiettivo del PID è dato dall'angolo di Eulero impostato dal telecomando, che viene confrontato con l'angolo di Eulero della posizione reale del drone (funzione presente nel file «`flight_control.c`»):
- L'uscita del PID è il valore della velocità dei 4 Motori (segnale PWM HW per ESC o Mosfet driving) per il quadricoptero:

```
FlightControlPID_OuterLoop(&euler_rc_fil,&euler_ahrs,&ahrs,&pid);
```

```
motor_pwm->motor1_pwm = motor_thr - pid->x_s2 - pid->y_s2 + pid->z_s2 + MOTOR_OFF1;
motor_pwm->motor2_pwm = motor_thr + pid->x_s2 - pid->y_s2 - pid->z_s2 + MOTOR_OFF2;
motor_pwm->motor3_pwm = motor_thr + pid->x_s2 + pid->y_s2 + pid->z_s2 + MOTOR_OFF3;
motor_pwm->motor4_pwm = motor_thr - pid->x_s2 + pid->y_s2 - pid->z_s2 + MOTOR_OFF4;
```

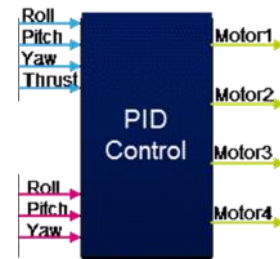
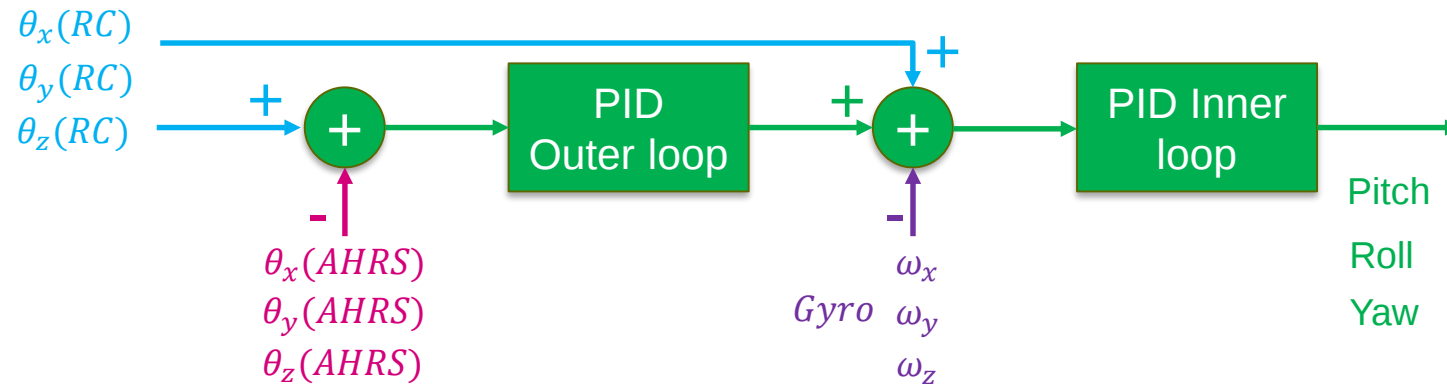
Controllo PID del Volo

53



Controllo PID del Volo

54



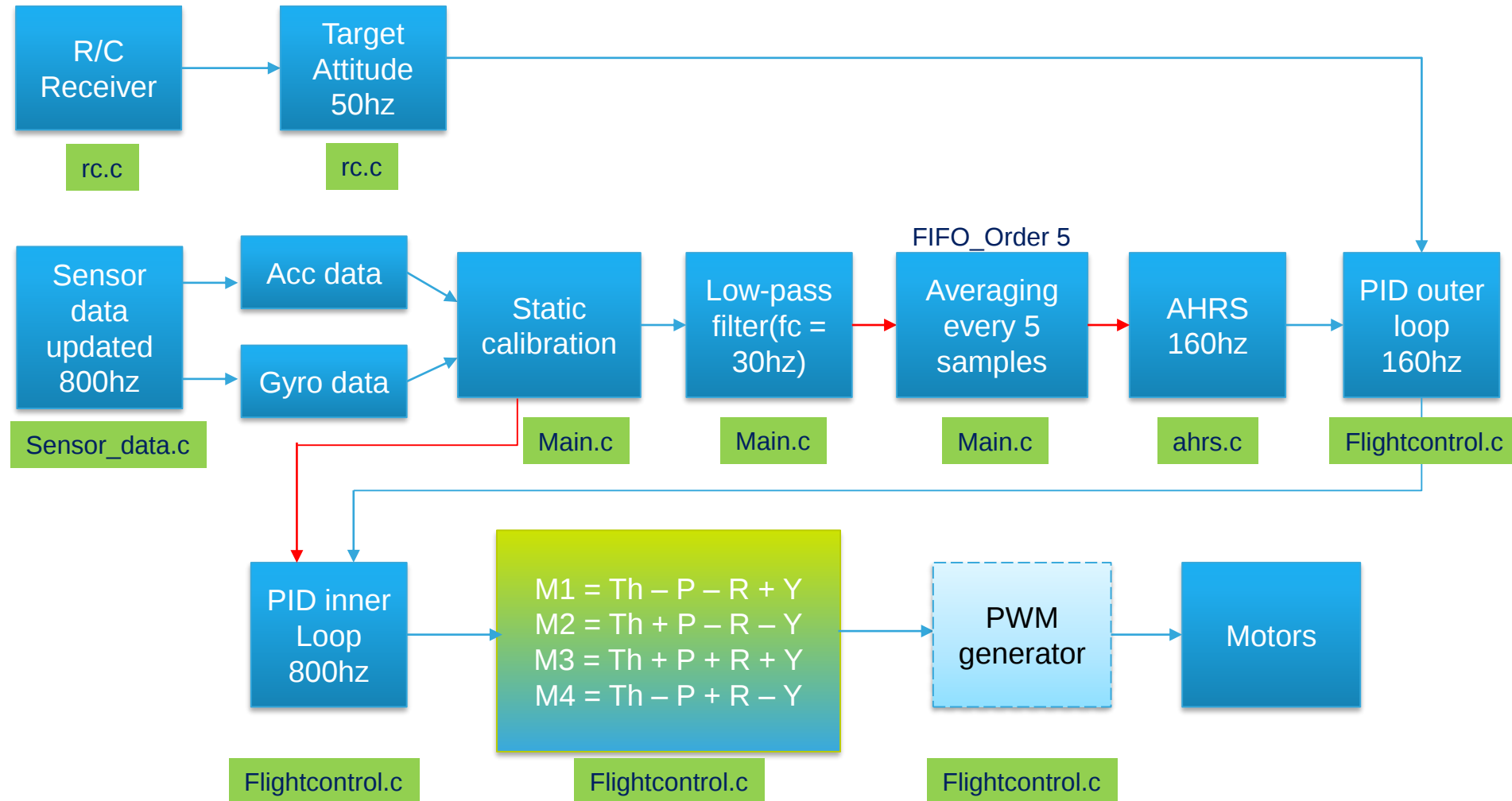
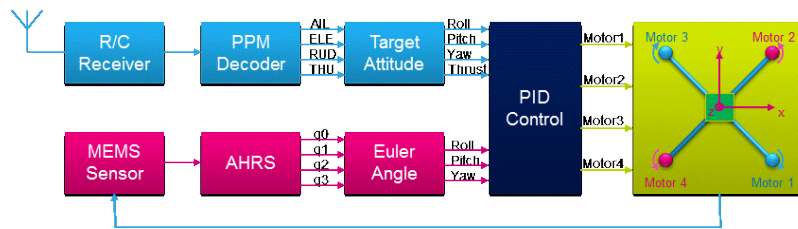
- In realtà il controllo PID viene eseguito in due fasi separate:
 - **PID Outer Loop:** confrontando gli angoli di Eulero bersaglio, dati dal telecomando, e gli angoli di Eulero del sistema AHRS, controlla l'angolo di inclinazione;
 - **PID Inner Loop:** usato per tracciare la velocità angolare.
- Facendo un modello matematico del sistema, è quasi impossibile rendere il sistema stabile con un singolo loop. L' *Inner Loop* deve essere necessariamente accoppiato al *Outer Loop* per stabilizzare il sistema.
- Le relative funzioni sono presenti nel file «flight_control.c»:

```
void FlightControlPID_OuterLoop(EulerAngleTypeDef *euler_rc, EulerAngleTypeDef *euler_ahrs,
AHRS_State_TypeDef *ahrs, P_PI_PIDControlTypeDef *pid);
```

```
void FlightControlPID_innerLoop(EulerAngleTypeDef *euler_rc, Gyro_Rad *gyro_rad,
AHRS_State_TypeDef *ahrs, P_PI_PIDControlTypeDef *pid, motorControlTypeDef *motor_pwm);
```

Schema del flusso dati

55

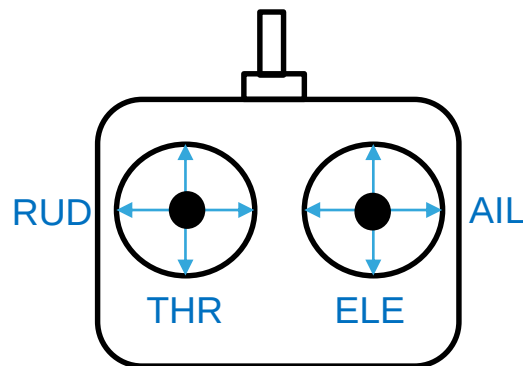




Appendice

Messa a Punto e Personalizzazioni

Calibrazione R/C (1/2) 57



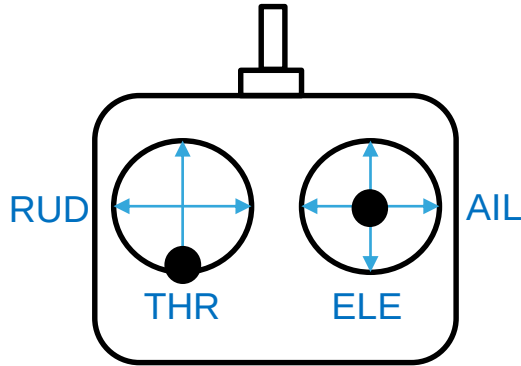
Dopo aver collegato un nuovo ricevitore del telecomando alla FCU, una delle prime operazioni da fare è controllare se i canali sono collegati correttamente e quindi fare una calibrazione dell'offset per ciascun canale.

Channel	Control	GPIO
CH1	AIL	PA0
CH2	ELE	PA1
CH3	THR	PA2
CH4	RUD	PA3

(*) Alcuni telecomandi consentono anche di programmare l'assegnazione dei canali.

- 1) Collegamento TX e RX del telecomando:** in un primo momento è meglio assicurarsi che l'operazione di collegamento TX e RX abbia avuto esito positivo, altrimenti nessun segnale verrà generato dal ricevitore RX alla FCU. Per questa operazione, fare riferimento alle istruzioni del produttore del telecomando.
- 2) Collegamento dei Canali :** dopo aver collegato il ricevitore RX alla FCU è necessario verificare che i collegamenti dei canali siano corretti. Spostando il joystick, verificare che THR, RUD, etc., siano in linea con le variabili `gAIL`, `gELE`, `gTHR`, `gRUD` (in «`rc.c`»). Nel caso di connessioni errate è possibile modificare la connessione HW (*) o SW nella funzione:

```
void update_rc_data(int32_t idx)
```

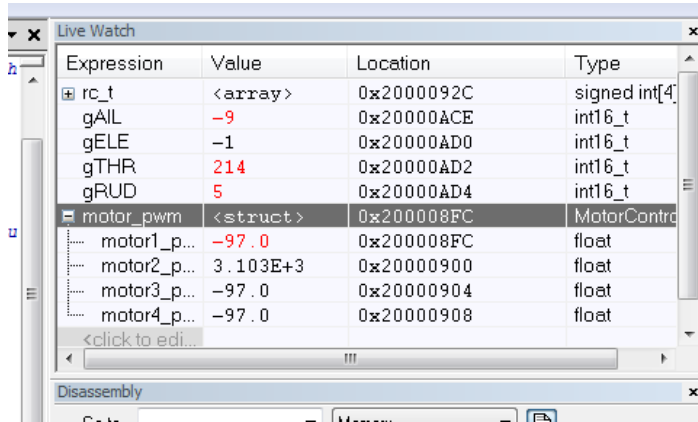


- 3) Calibrazione degli offset dei canali:** è necessario regolare l'offset per ciascun canale (*).
- Posizionare il joystick sinistro al centro in basso.
 - Posizionare il joystick destro al centro
 - Prendere nota dei valori delle variabili gAIL, gELE, gTHR, gRUD (in «rc.c»).
Se il telecomando è calibrato correttamente, tutti valori dovrebbero essere *vicino allo zero*.
 - Se i valori sono diversi da zero, è necessario regolare l'offset modificando il valore delle seguenti costanti (in «rc.h») :

```
#define AIL_MIDDLE 6269  
#define ELE_MIDDLE 6051  
#define RUD_MIDDLE 5949  
#define THR_BOTTOM 4437
```

Controllo dei collegamenti Motori

59



Expression	Value	Location	Type
rc_t	<array>	0x2000092C	signed int[4]
gAIL	-9	0x20000ACE	int16_t
gELE	-1	0x20000AD0	int16_t
gTHR	214	0x20000AD2	int16_t
gRUD	5	0x20000AD4	int16_t
motor_pwm	<struct>	0x200008FC	MotorContro
motor1_p...	-97.0	0x200008FC	float
motor2_p...	3.103E+3	0x20000900	float
motor3_p...	-97.0	0x20000904	float
motor4_p...	-97.0	0x20000908	float

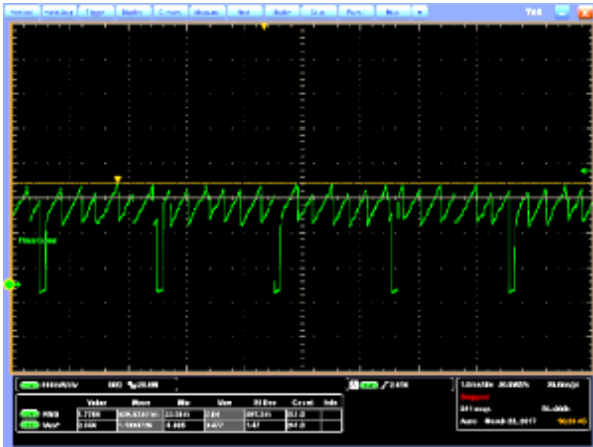
Condizione di collegamento errato.



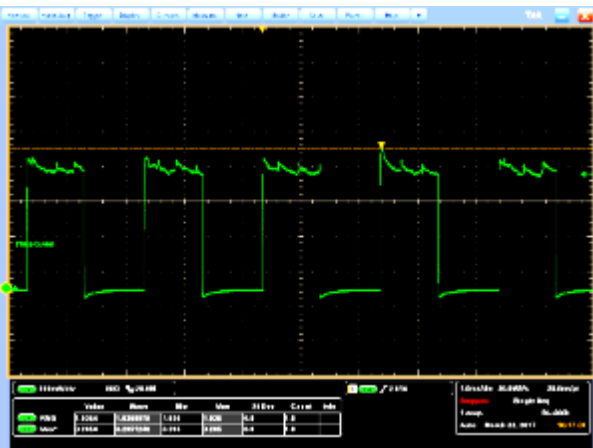
- Potrebbe essere facile commettere un errore nella connessione dei Motori alla FCU (configurazione corretta descritta nella sezione «[Passo 2 - Collegamento Dei Motori](#)»).
- È possibile verificarlo facilmente aumentando il valore di THR (Spinta) (quanto basta per far girare il motore ma non per sollevare il quadricottero) e forzare un'inclinazione. Nella figura, viene mostrato cosa succede **in caso di connessione errata**: il motore nella posizione inferiore viene fermato mentre quello in posizione più alta ruota >> dovrebbe essere esattamente l'opposto per stabilizzare il drone.

Configurazione Motori : Corrente ai Mosfet

60



**Corrente
Motore
Spinta Max
($R_g = 1K\Omega$)**



**Corrente
Motore
Spinta Max
($R_g = 0\Omega$)**

- Potrebbe essere necessario sostituire la resistenza di gate (R_g) dei Mosfet in caso di motore DC a spazzola.
- Nel caso in cui R_g è troppo grande, il Mosfet potrebbe non accendersi correttamente e la corrente disponibile per il motore potrebbe non essere sufficiente per sollevare il quadricottero. Allo stesso modo, la carica necessaria per accendere il MOS è legata alla carica del GPIO del STM32.
- Si noti che la corrente necessaria può variare in base al motore, alla dimensione dell'elica e al peso complessivo del drone. Per un piccolo telaio come nella figura è stato selezionato $R_g=100\Omega$ come giusto compromesso.

Messa a Punto della Stabilizzazione PID

61

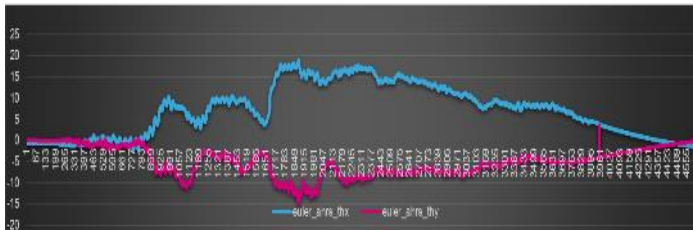
- Per ogni configurazione di quadricottero telaio / motori / FCU, è necessario mettere a punto i parametri PID (nel file «flight_control.h») per trovare le impostazioni migliori che fanno volare il drone con una buona stabilità e reattività.
 - Inizialmente bisogna controllare che il drone stia volando in condizione di «volo stazionario» senza oscillazioni (solo Throttle applicato da telecomando).
1. Iniziare con un valore basso di **KP** (lo stesso valore per gli assi x e y se il design del quadricottero è simmetrico e ben bilanciato meccanicamente, come nella maggior parte dei casi) e un valore inferiore di **KI**.
 2. Aumentare **KP** senza avere oscillazioni.
 3. Aumentare il valore di **KI** senza avere oscillazioni.
 4. Si consiglia di mantenere lo stesso valore PID per Roll e Pitch nella maggior parte dei casi.
 5. Nella maggior parte dei casi (eccetto per i droni da corsa in cui è necessaria una risposta immediata) è meglio mantenere il valore **KD** a zero per garantire una buona stabilità.

```
#define ROLL_PID_KP1      3
#define ROLL_PID_KI1      0
//#define ROLL_PID_KP2    800    /* default */
#define ROLL_PID_KP2      100    /* test minidrone */
//#define ROLL_PID_KI2    400    /* default */
#define ROLL_PID_KI2      100    /* test minidrone */
#define ROLL_PID_KD2      10
...

#define PITCH_PID_KP1     ROLL_PID_KP1
#define PITCH_PID_KI1     ROLL_PID_KI1
#define PITCH_PID_KP2     ROLL_PID_KP2
#define PITCH_PID_KI2     ROLL_PID_KI2
#define PITCH_PID_KD2     ROLL_PID_KD2
```

Datalog mediante porta UART

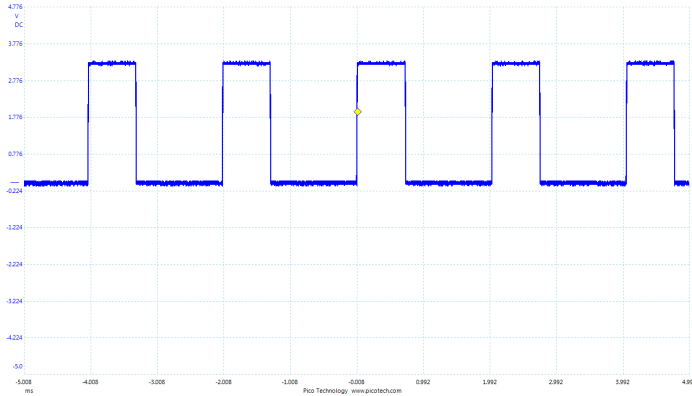
62



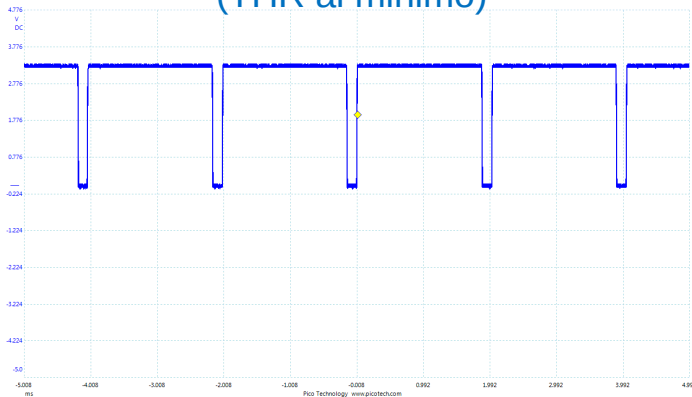
- Durante la fase di test è molto utile creare un registro dati e tracciare diverse variabili, come gli input del controllo remoto, i dati raw dei MEMS, angoli di Eulero, i PWM dei motori, etc.
- Il modo più semplice per eseguire questo test è:
 - Connettere un modulo Bluetooth con protocollo SPP alla porta UART e ottenere dati su PC con un'applicazione Console / Terminale (con SPP il modulo BLT sarà visto come una porta COM virtuale).
 - Basterà quindi utilizzare delle PRINTF per visualizzare i dati sul PC, ma è meglio prestare attenzione a non introdurre troppo ritardo nel *main*.
- Un'altra opzione consiste nell'utilizzare il modulo BLE a bordo, ma l'utente dovrebbe occuparsi da solo di scrivere un semplice protocollo per il datalog.

Messa a punto FW per ESC Esterno

63



PWM su P1, P2, P4, P5
(THR al minimo)



PWM su P1, P2, P4, P5
(THR al massimo)

- Nel caso di ESC esterno, possono essere necessarie alcune modifiche per assicurarsi che il segnale di uscita PWM, per ciascun motore, sia compatibile con il segnale di ingresso ESC.
- La maggior parte degli ESC accetta un segnale in ingresso con una frequenza variabile da 50 a 400 Hz e il tempo T_{on} del PWM è proporzionale alla velocità del motore. Deve essere controllato anche il tempo di T_{on} minimo per il quale l'ESC è armato (quando non armato di solito l'ESC emette un particolare bip di allarme).
- Potrebbe essere necessario modificare le seguenti variabili:
 - MIN_THR ([flight_control.h](#))
 - rotor_MIN_PWM_VALUE ([rotor.h](#))
 - rotor_MAX_PWM_VALUE ([rotor.h](#))
 - htim4.Init.Prescaler ([main.c](#))

- Nelle impostazioni predefinite il valore di PITCH e di ROLL massimo è 20 gradi, mentre il YAW è limitato alla velocità massima di rotazione di 100 deg/s. Questi valori sono buoni per i principianti o per acquisire confidenza con il quadricottero.
- Per piloti più esperti (e per corse / uso sportivo), è possibile aumentare questi valori nel file «rc.h»:

```
// Maximum roll/pitch 20deg
#define PITCH_MAX_DEG    20
#define ROLL_MAX_DEG     20
#define YAW_MAX_DEG      (60.0*SENSOR_SAMPLING_TIME)
#define YAW_MIN_RAD      0.0872
#define EULER_Z_TH       600
```

- STEVAL-FCU001-V1:

- http://www.st.com/content/st_com/en/products/evaluation-tools/solution-evaluation-tools/sensor-solution-eval-boards/steval-fcu001v1.html

- Documentation

- DIY Drones: <http://diydrones.com/>
- Unmanned Aircraft Systems - Federal Aviation Administration: <https://www.faa.gov/uas/>
- Quaternion-Based Kalman Filter for AHRS Using an Adaptive-Step Gradient Descent Algorithm: <http://journals.sagepub.com/doi/full/10.5772/61313>
- Filtro Complementare – Metodo di Mahony: <http://ieeexplore.ieee.org/document/4608934/>
- IMU Data Fusing: Complementary, Kalman, and Mahony Filter: <http://www.olliv.eu/2013/imu-data-fusing/>
- Quadcopter Pid Explained: <https://oscarliang.com/quadcopter-pid-explained-tuning/>
- PID Tuning and Quadcopter Flight Behaviour: <https://oscarliang.com/understanding-pid-for-quadcopter-rc-flight/>

- Datasheets:

- STM32F401 : <http://www.st.com/en/microcontrollers/stm32f401.html?querycriteria=productId=LN1810>
- LSM6DSL: <http://www.st.com/en/mems-and-sensors/lsm6dsl.html>
- LPS22HD: http://www.st.com/content/st_com/en/products/mems-and-sensors/pressure-sensors/lps22hd.html
- LIS2MDL: http://www.st.com/content/st_com/en/products/mems-and-sensors/e-compasses/lis2mdl.html
- SPBTLE-RF: http://www.st.com/content/st_com/en/products/wireless-connectivity/bluetooth-low-energy/spbtle-rf.html
- STL6N3LLH6: http://www.st.com/content/st_com/en/products/power-transistors/power-mosfets/n-channel-stripfet-12-v-to-30-v/stl6n3llh6.html
- STC4054: http://www.st.com/content/st_com/en/products/power-management/battery-management-ics/battery-chargers/stc4054.html
- LD39015: http://www.st.com/content/st_com/en/products/power-management/linear-voltage-regulators/low-dropout-ldo-linear-regulators/low-iq-ldo-regulators/ld39015.html



life.augmented

Thank You

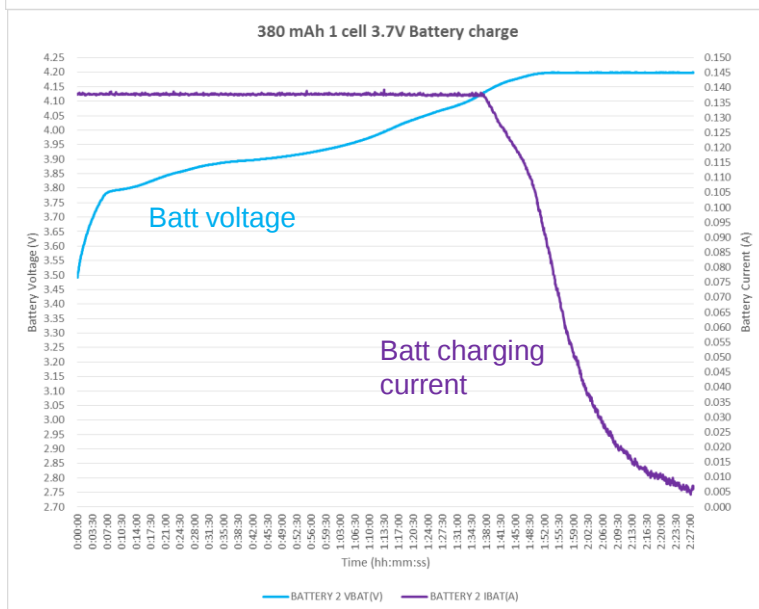
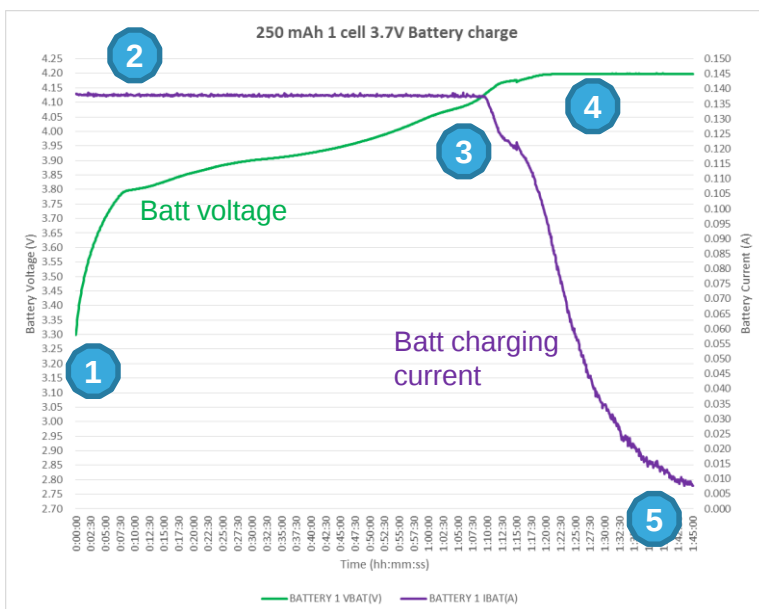


Appendice

Caricabatteria integrato per Batterie LiPO

Carica della Batteria LiPO

68



Nello schema è mostrata la tipica curva di carica per una batteria LiPo. Consiste dei seguenti passaggi:

- 1 Batteria scarica alla tensione di interruzione (2,7 V).
- 2 Corrente di carica impostata su ~ 135 mA (*).
- 3 Quando la tensione della batteria raggiunge ~ 4,1 V, la ricarica passa dalla modalità corrente costante alla modalità a tensione costante. La tensione della batteria aumenta lentamente fino a 4,2 V e la corrente di carica diminuisce.
- 4 Tensione di 4,2 V alla massima carica
- 5 La corrente di carica ha raggiunto il valore finale (1/10 della corrente di carica), mantiene solo la carica della batteria.

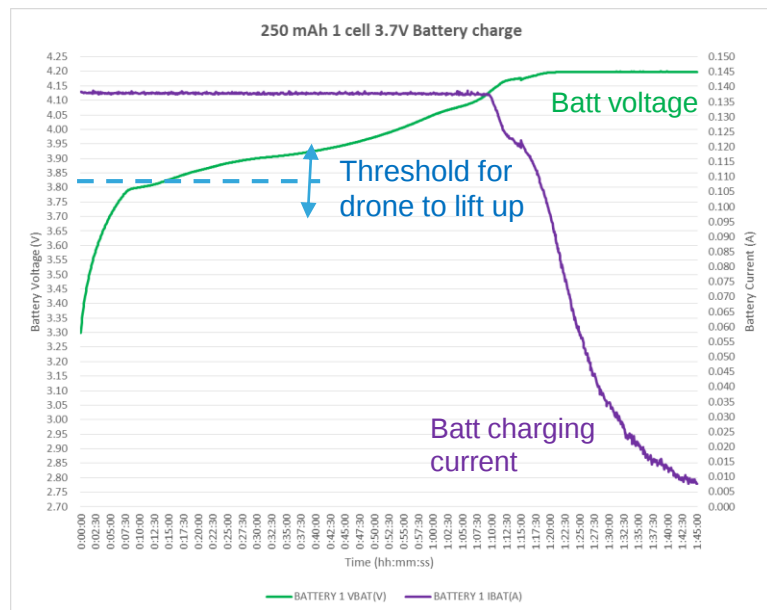
I due diagrammi sono relativi a 2 batterie diverse:

- Batteria 250mAh >> ~1h 25m da completamente scarica a carica completa
- Batteria 380mAh >> ~2h da completamente scarica a carica completa

(*) La corrente di carica può essere modificata dalla resistenza collegata al pin PROG di STLC4054.

Uso della Batteria LiPO nel Drone

69



- È importante prendere nota dei seguenti suggerimenti quando si utilizza una batteria LiPO:
 1. A seconda di diversi fattori (il peso totale del drone, i motori e le eliche utilizzati, ...) il drone potrebbe non essere in grado di sollevarsi (o librarsi a pochi cm da terra) già a una tensione superiore al valore di scarica completa (ovvero 3,5 V o 3.6V). Sarà necessario caricare la batteria da questo livello di voltaggio, ma ridurrà il tempo di ricarica della batteria.
 2. La scelta di una batteria di maggiore capacità aumenterà il tempo di volo, ma bisogna prestare attenzione al peso totale e cercare il giusto compromesso.
 3. Dopo il volo, la batteria dovrebbe essere immediatamente scollegata dalla scheda FCU. Se rimane collegato per lungo tempo e la tensione della batteria scende sotto i 2,7 V, c'è il rischio di danneggiare in modo permanente la batteria. Non sarà più possibile ricaricarla.
 4. È raccomandato usare una corrente di carica non superiore a ~ 1/3 della capacità massima (per esempio: 380 mAh -> corrente di carica 130 mA).



380mAh 1 cell Batt

250mAh 1 cell Batt



Appendice

Nozioni Basi del Protocollo BLE

Cos'è il Bluetooth Smart (Low Energy)

- Bluetooth Smart è uno standard globale, con una tecnologia di alimentazione molto bassa
- I dispositivi Bluetooth Smart consumano solo una piccola parte della potenza dei classici prodotti Bluetooth (v3.0)
- La tecnologia Bluetooth Low Energy consente ai dispositivi dotati di batterie a bottone di essere collegati in modalità wireless a dispositivi abilitati Bluetooth standard.
 - Bassa corrente di picco (ad esempio $<20\text{ mA}$)
 - Bassa corrente media (ad es. $<100\text{ uA}$)
- Ciò consente di utilizzare questa tecnologia su un'ampia gamma di applicazioni di sensori che trasmettono piccole quantità di dati.
 - Settore automobilistico
 - Sport e fitness
 - Assistenza sanitaria
 - Intrattenimento
 - Domotica
 - Sicurezza e prossimità



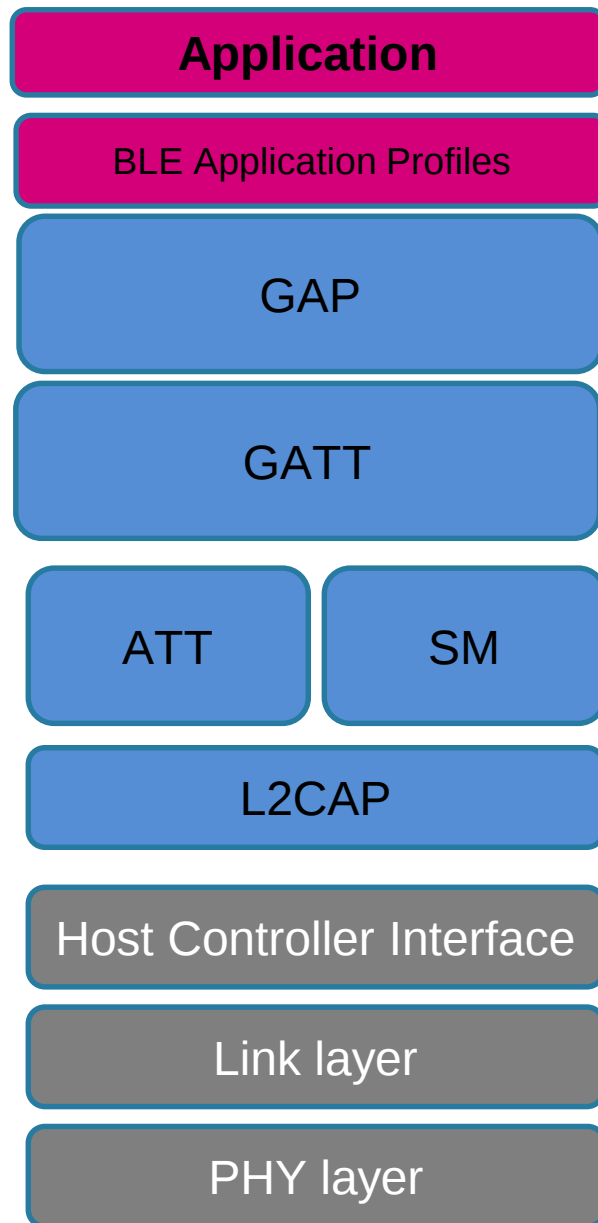
BLE: confronto con il classico Bluetooth

72

Caratteristiche	BR/EDR	LE (4.0)	Notes
RF Channels	79	40	2 MHz spacing in LE
Modulation	GFSK	GFSK	Simple and effective
Modulation index	0.25 to 0.35	0.45 to 0.55	Wider signal - more robust
Max TX power	+20 dBm (class 1) +4 dBm (class 2)	+10 dBm	No “class” structure +10dBm regulatory limit
Packet Format	6	2	ID, FHS, DM, DH, 2-DH, 3-DH - Advertising / Data
Ack Packet Len	126 μ s	80 μ s	63% shorter
Max packet Size	2875 μ s = 1021 octets	328 μ s = 27 octets	LE very short L2CAP payload
CRC	16	24	LE stronger
Encryption	Safer+	AES-128	LE stronger
Topology	flexible	Pure star	LE simpler

Stack del Protocollo BLE

73

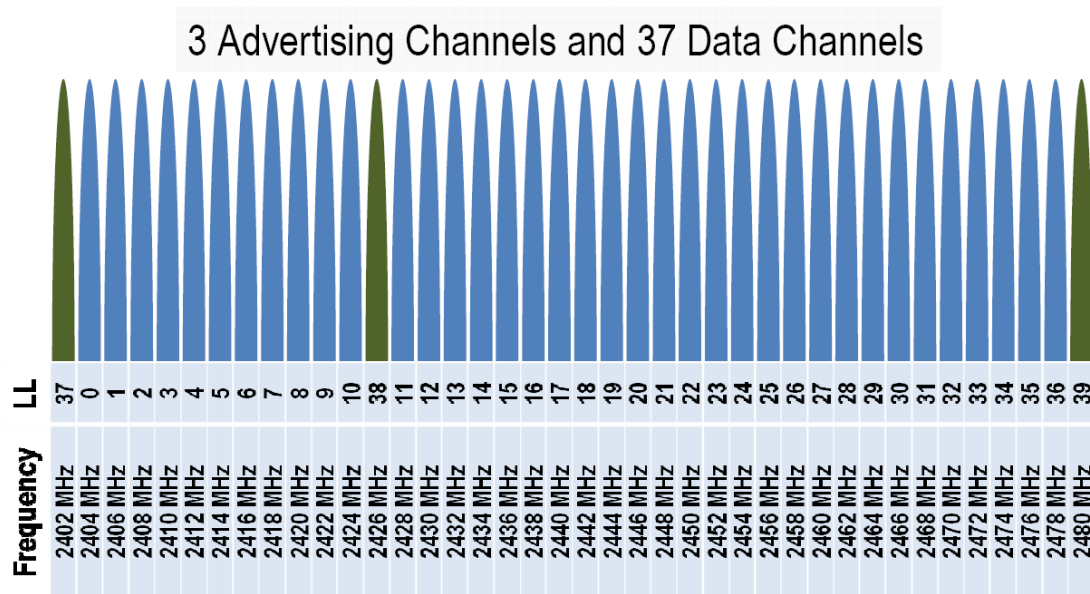


- Profilo di accesso generico (Generic Access Profile - GAP)
 - Definisce in che modo i dispositivi possono individuare e connettersi tra loro e in che modo si collegano
- Profilo Generico degli attributi (Generic Attribute Profile - GATT)
 - Definisce i modi standard in cui i servizi, le caratteristiche e i descrittori possono essere individuati e quindi utilizzati.
- Protocollo degli attributi (Attribute Protocol - ATT)
 - È un protocollo per la individuazione, la lettura e la scrittura di attributi su un dispositivo accoppiato.
- Security Manager (SM)
 - Gestisce la comunicazione sicura
- L2CAP
 - Multiplexer di protocollo
 - Segmentazione e riassemblaggio di pacchetti
- HCI
 - Interfaccia tra host e controller
- Link Layer
 - Gestisce pacchetti, canali, advertising, scansioni e connessioni
- Physical Layer
 - Transmette e riceve bit

Stack del Protocollo BLE: Physical Layer - PHY

74

- 2.4 GHz ISM Ricetrasmittente a banda
 - Suddiviso in 40 Canali RF
 - 2 MHz Spaziatura dei canali.
 - Due tipi di canali (3 Advertising e 37 Dati)



- Modulazione GFSK
- 2.4 GHz ISM Ricetrasmittente a banda (BT)
 - BT = 0.5
 - Modulation Index = 0.5



GFSK:

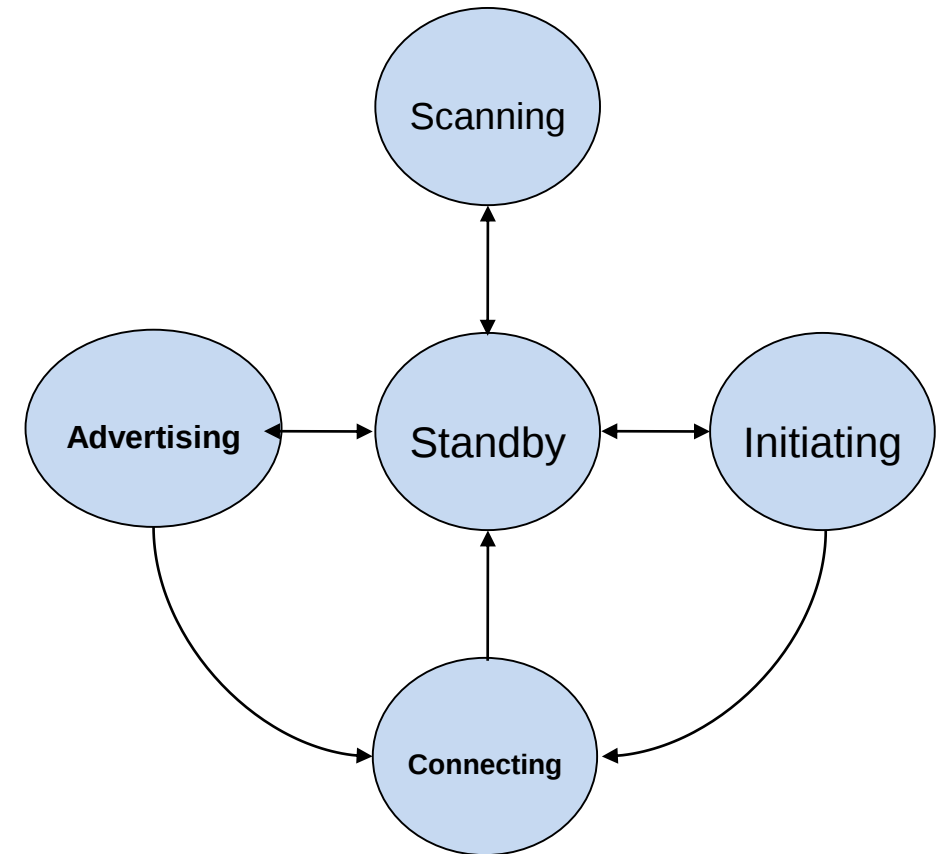
“pulse shaping”

Il filtro gaussiano uniforma le transizioni da zero a uno per ridurre la larghezza spettrale

Stack del Protocollo BLE: Link Layer States

75

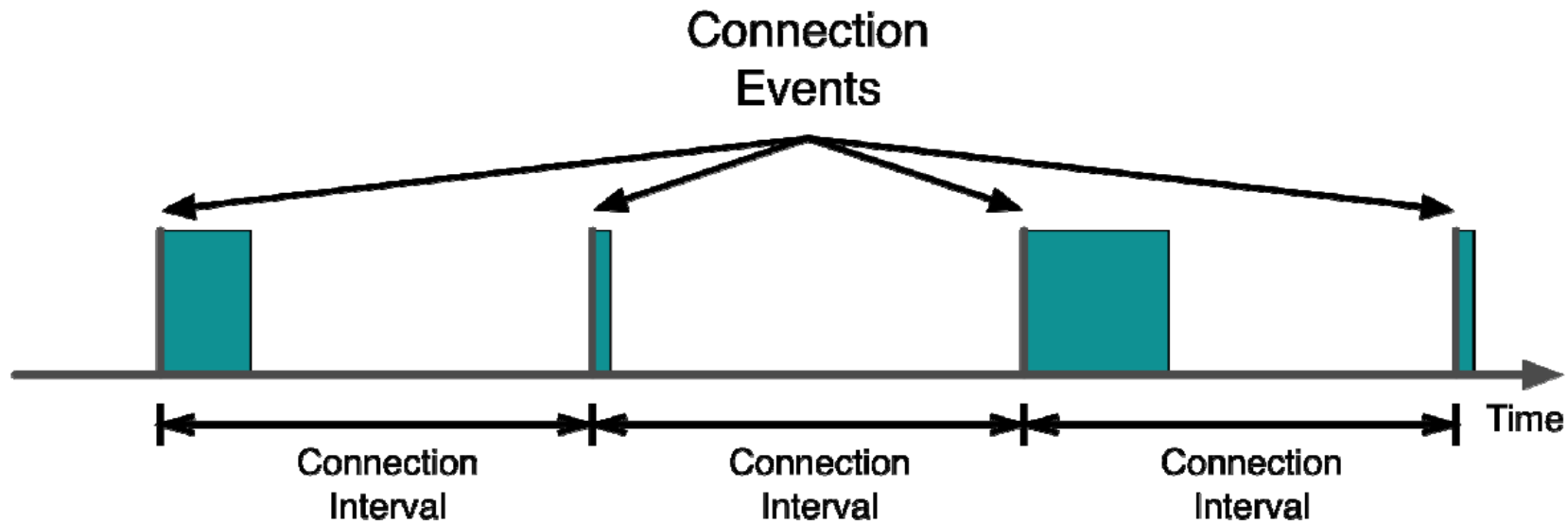
- Standby
 - Non trasmette ne riceve dati
- Advertising
 - Trasmette segnali sui canali advertising
- Scanning
- Ricerca advertisers
- Initiating
 - Inizializza la connessione al advertiser
- Connection
 - Il dispositivo Iniziatore è in Master Role
 - Comunica con I dispositivi in Slave Role, definisce il timing delle trasmissioni
 - Il dispositivo Advertiser è in Slave Role
 - Comunica con un singolo dispositivo in Master Role



Stack del Protocollo BLE: Link Layer

76

- Il Master trasmette periodicamente al evento di connessioni
 - Connection Interval sent in `CONNECT_REQ`
- All'aumentare dei bit di dati gli eventi di connessione vengono estesi automaticamente



Stack del Protocollo BLE: HCI

77

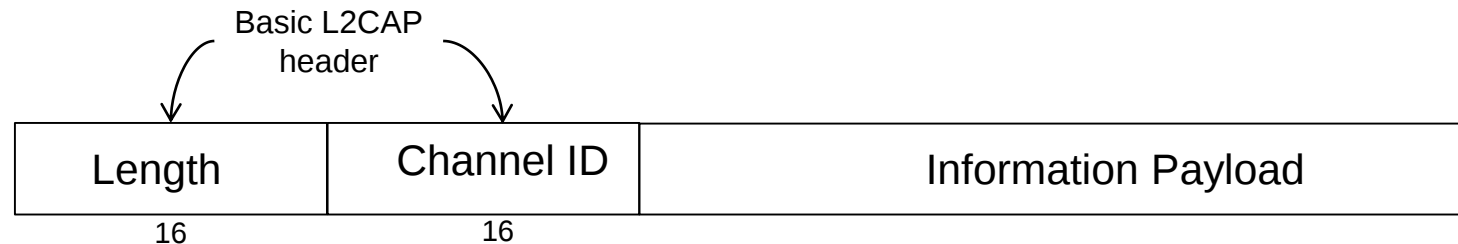
- La separazione tra host e controller è considerata come nei dispositivi Bluetooth standard. L'interfaccia HCI può riutilizzare il comando LE esistente per aggiungere nuove funzionalità.
- HCI fornisce i mezzi di comunicazione tra l'host e il controller. Può essere inoltre implementato:
 - Attraverso l'interfaccia hardware (SPI, UART, USB)
 - o
 - Attraverso l'API del software

Stack del Protocollo BLE: L2CAP

78

- Fornisce servizi di frammentazione e multiplexing di protocolli/canali per i protocolli di livello superiore.
- Un identificatore di canale (CID) è il nome locale che rappresenta un end-point del canale logico sul dispositivo. Tutte le applicazioni Bluetooth Low Energy utilizzano il CID fisso

L2CAP PDU



CID	Description	Notes
0x0004	Attribute Protocol	Fixed channel for accessing device attributes
0x0005	L2CAP LE Signaling	Fixed channel to manage <i>Bluetooth</i> low energy devices
0x0006	Security Manager Protocol	Fixed channel to enable security protocols

Stack del Protocollo BLE: SM

79

- Definisce il protocollo per impostare un collegamento sicuro
- Esegue autenticazione e gestione delle chiavi
 - Accoppiamento e distribuzione delle chiavi: «Mi fido di questo dispositivo, ed ecco una chiave per dimostrarlo».
- Utilizza AES-128 come algoritmo di crittografia per le procedure di sicurezza.

Stack del Protocollo BLE: Definizioni SM

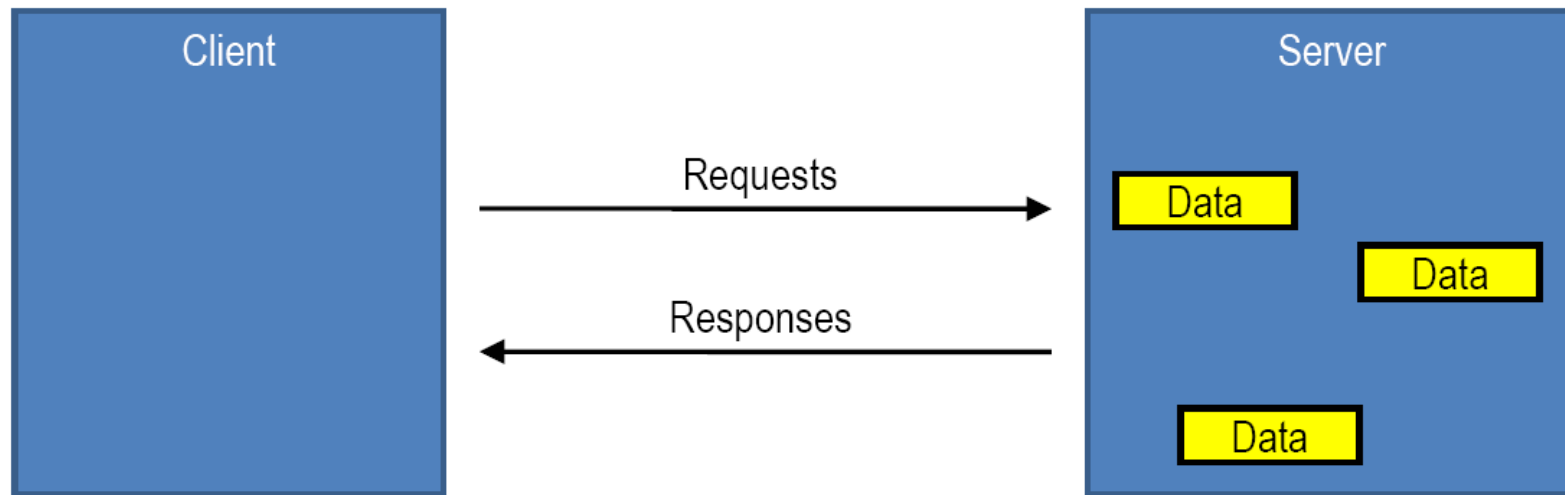
80

- Pairing
 - Una procedura che autentica due dispositivi e crea una chiave di collegamento comune che può essere utilizzata come base per una relazione di fiducia o una (singola) connessione sicura
- Bonding
 - Crea una relazione tra due dispositivi Bluetooth basati su una chiave di collegamento comune (un legame (bond)). La chiave di collegamento viene creata e scambiata (pairing) durante la procedura di bonding e dovrebbe essere archiviata da entrambi i dispositivi Bluetooth, **per essere utilizzata nelle autenticazioni successive.**
- Out Of Band (OOB) Data
 - Dati scambiati in un diverso sistema di comunicazione (ad esempio NFC).
- MITM (Man-In-The-Middle) protection
 - Protezione contro gli intercettatori.

Stack del Protocollo BLE: Architettura Client/Server

81

- I server hanno dati, i client vogliono usare questi dati.
- I server espongono i dati utilizzando gli attributi:



Stack del Protocollo BLE: Caratteristiche

82

- Una caratteristica è un valore di base utilizzato per lo scambio tra server e client (es: temperatura, stato della batteria, nome del dispositivo).
- Una caratteristica è un valore utilizzato in un servizio, insieme a proprietà e informazioni di configurazione, su come si accede al valore e su come il valore viene visualizzato o rappresentato.
- Una definizione di caratteristica contiene
 - Dichirazione della caratteristica,
 - Proprietà della caratteristica
 - Un valore
 - Può anche contenere descrittori che rappresentano il valore o consentono la configurazione del server rispetto al valore della caratteristica.

Attribute Handle	Attribute Type s	Attribute Value			Attribute Permissions
0xNNNN	0x2803—UUID for «Characteristic»	Character-istic Proper-ties	Character-istic Value Attribute Handle	Characteris-tic UUID	Read Only, No Authentication, No Authorization

- Proprietà della caratteristica: leggere, scrivere, scrivere senza risposta, notificare, indicare, ...
- UUID della caratteristica: UID Bluetooth a 16 bit o UUID a 128 bit per un valore di caratteristica.

Stack del Protocollo BLE: Valore della Caratteristica

83

- La dichiarazione «Characteristic Value» contiene il valore della caratteristica.
- È il primo attributo dopo la dichiarazione della caratteristica.

Attribute Handle	Attribute Type	Attribute Value	Attribute Permissions
0xNNNN	0xuuuu – 16-bit Bluetooth UUID or 128-bit UUID for Characteristic UUID	Characteristic Value	Higher layer profile or implementation specific

Characteristic value declaration

- Proprietà della caratteristica: leggere, scrivere, scrivere senza risposta, notificare, indicare, ...
- UUID della caratteristica: UID Bluetooth a 16 bit o UUID a 128 bit per un valore di caratteristica

Stack del Protocollo BLE: Servizio

84

- Un servizio è una raccolta di caratteristiche che operano insieme per rendere un servizio globale a un profilo applicativo (ad esempio: il servizio Health Thermometer include le caratteristiche per un valore di misurazione della temperatura e un intervallo di tempo tra le misurazioni).
- Può fare riferimento ad altri servizi (servizi secondari).
- Una definizione di servizio deve contenere una dichiarazione di servizio e può contenere definizioni di include e definizioni di caratteristiche

Attribute Handle	Attribute Type	Attribute Value	Attribute Permission
0xNNNN	0x2800 – UUID for «Primary Service» OR 0x2801 for «Secondary Service»]	16-bit Bluetooth UUID or 128-bit UUID for Service	Read Only, No Authentication, No Authorization

Service declaration

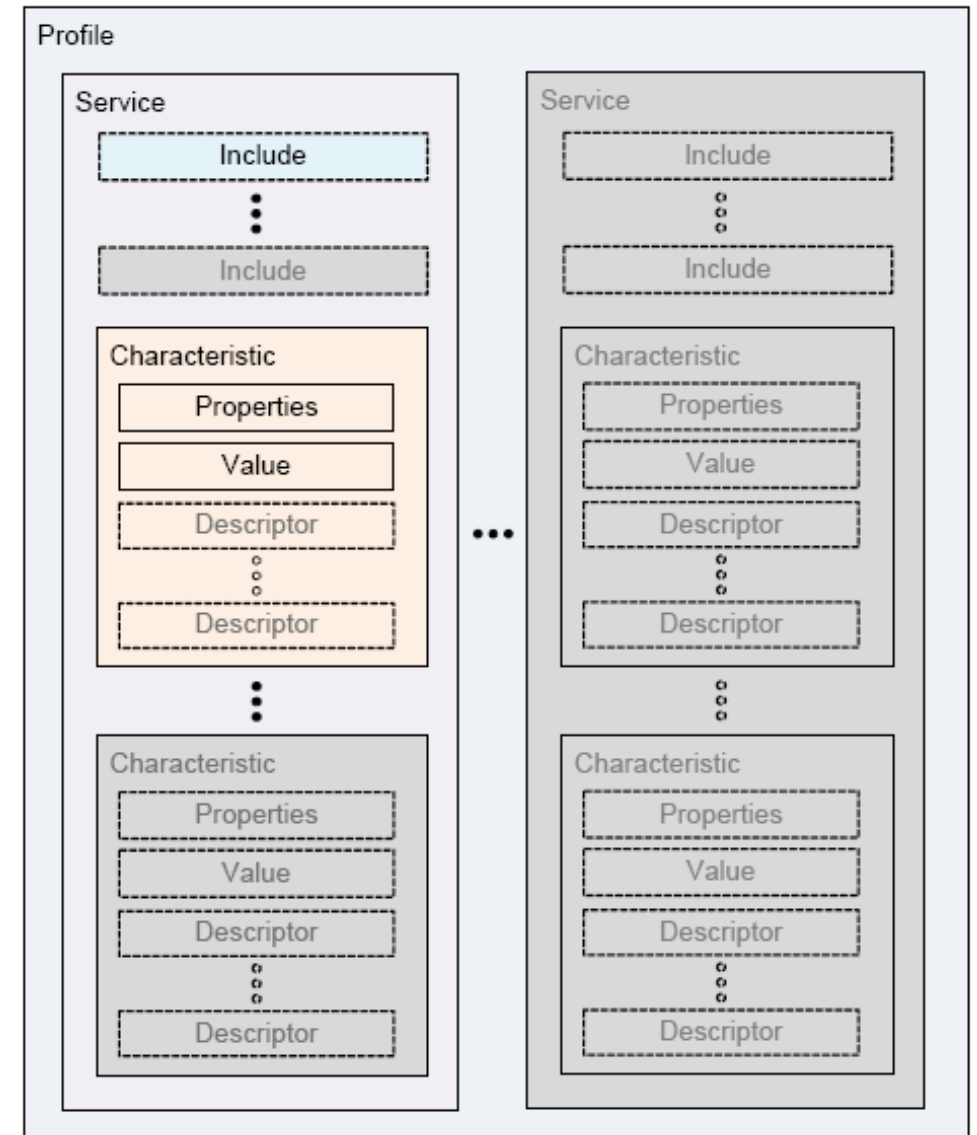
Attribute Handle	Attribute Type	Attribute Value			Attribute Permission
0xNNNN	0x2802 – UUID for «Include»	Included Service Attribute Handle	End Group Handle	Service UUID	Read Only, No Authentication, No Authorization

Include declaration

Stack del Protocollo BLE: Profilo GATT

85

- Il profilo GATT specifica la struttura in cui vengono scambiati i dati.
- Definisce i modi standard in cui i servizi, le caratteristiche e i descrittori possono essere scoperti e utilizzati.
- Un profilo è composto da uno o più servizi richiesti per affrontare un caso d'uso.
- Un servizio è composto da caratteristiche o riferimenti ad altri servizi.
- Ogni caratteristica contiene un valore e può contenere informazioni opzionali sul valore.
- Il servizio e le caratteristiche contengono i dati del profilo e sono tutti memorizzati in Attributi sul server



Stack del Protocollo BLE: Sensor Profile GATT database

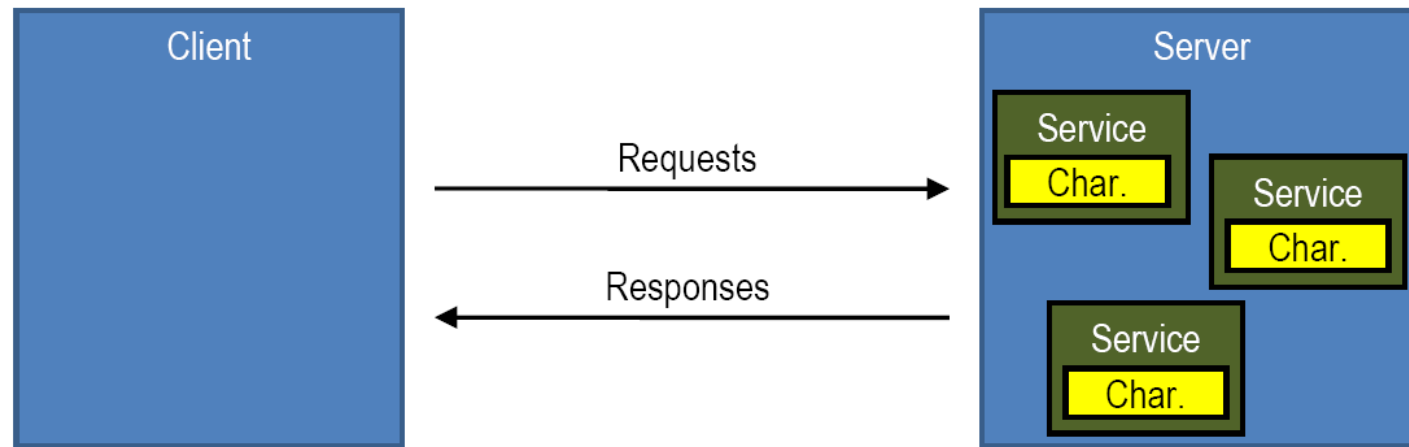
86

# Handle	UUID (16 or 128bit)	Attribute Type	Properties	Initial Parameter Value
			W R R E N S B R R N S W O N G w x D D O P R T D N R T	
1	0001	2800	Primary Service	{Service=0x1801 ("Attribute Profile")}
2	0002	2803	Characteristic	X {handle=0x0003, UUID=0x2A05}
3	0003	2A05	Service Changed	{start handle=0x0001, end handle=0xFFFF}
4	0004	2902	Client Characteristic Configuration	0x0000
5	0005	2800	Primary Service	{Service=0x1800 ("Generic Access Profile")}
6	0006	2803	Characteristic	X X X X {handle=0x0007, UUID=0x2A00}
7	0007	2A00	Device Name	"bluenrg"
8	0008	2803	Characteristic	X X X {handle=0x0009, UUID=0x2A01}
9	0009	2A01	Appearance	0x0000
12	000C	2800	Primary Service	{Service=0x02366E80CF3A11E19AB40002A5D5C51B ("Acc Service")}
13	000D	2803	Characteristic	X {handle=0x000E, UUID=0xE23E78A0CF4A11E18FFC0002A5D5C51B}
14	000E	E23E78A0CF4A11E18FFC0002A5D5C51B	Free Fall	0x00
15	000F	2902	Client Characteristic Configuration	0x0000
16	0010	2803	Characteristic	X X {handle=0x0011, UUID=0x340A1B80CF4B11E1AC360002A5D5C51B}
17	0011	340A1B80CF4B11E1AC360002A5D5C51B	Acceleration	0x000000000000
18	0012	2902	Client Characteristic Configuration	0x0000
19	0013	2800	Primary Service	{Service=0x02366E80CF3A11E19AB40002A5D5C51B ("Env Service")}
20	0014	2803	Characteristic	X {handle=0x0015, UUID=0xA32E5520E47711E2A9E30002A5D5C51B}
21	0015	A32E5520E47711E2A9E30002A5D5C51B	Temperature	0x0000
22	0016	2904	Characteristic Format	{format=0x0E, exp=-1, unit=0x272F, n_sp=0x00, descr=0x0000}
23	0017	2803	Characteristic	X {handle=0x0018, UUID=0xCD20C480E48B11E2840B0002A5D5C51B}
24	0018	CD20C480E48B11E2840B0002A5D5C51B	Pressure	0x000000
25	0019	2904	Characteristic Format	{format=0x0F, exp=-5, unit=0x2780, n_sp=0x00, descr=0x0000}
26	001A	2803	Characteristic	X {handle=0x001B, UUID=0x01C50B60E48C11E2A0730002A5D5C51B}
27	001B	01C50B60E48C11E2A0730002A5D5C51B	Humidity	0x0000
28	001C	2904	Characteristic Format	{format=0x06, exp=-1, unit=0x2700, n_sp=0x00, descr=0x0000}

Stack del Protocollo BLE: GATT

87

- Definisce un framework di servizi utilizzando il protocollo degli attributi.
- Il framework definisce:
 - Procedure (ad esempio, scoprire servizi o caratteristiche, attributi di lettura / scrittura).
 - Formato dei servizi e loro caratteristiche



Stessa architettura del server client del Protocollo degli Attributi tranne che i dati sono incapsulati in "Servizi" e i dati sono esposti in "Caratteristiche"