

Telemetria Didattica per APR: Quota Barometrica e Accelerazione Verticale acquisite via BLE con SensorTag CC2650

Prof. Ing. Biagio Raucci*

Anno Scolastico 2025–2026

Sommario

Nel contesto della didattica delle Costruzioni Aeronautiche e della strumentazione di bordo, la stima della quota e l’osservazione delle variazioni di moto verticale rappresentano due grandezze fondamentali per comprendere i principi di navigazione e controllo degli APR (Sistemi a Pilotaggio Remoto). Questo lavoro presenta un sistema di telemetria essenziale e robusto, progettato con finalità formative: acquisizione via Bluetooth Low Energy (BLE) dei dati provenienti da un nodo sensoriale Texas Instruments CC2650 SensorTag, visualizzazione in tempo reale della quota barometrica (modalità QFE) e dell’accelerazione verticale netta. Il software è sviluppato in Python con interfaccia grafica PyQt6 e grafici in tempo reale mediante pyqtgraph, includendo procedure operative tipiche dell’avionica: connessione, calibrazione a terra, pausa della telemetria e sincronizzazione temporale.

Indice

1	Introduzione e obiettivi didattici	3
2	Setup sperimentale e architettura del sistema	3
2.1	Nodo sensore: TI CC2650 SensorTag	3
2.2	Stazione di controllo a terra	4
3	Comunicazione BLE e acquisizione dati	4
3.1	Principio di funzionamento BLE: notifiche GATT	4
3.2	Decodifica della pressione	5
3.3	Decodifica dell’accelerazione lungo z	5
4	Altimetria barometrica in modalità QFE	5
4.1	Idea fisica: pressione come “altezza”	5
4.2	Formula utilizzata	5
4.3	Osservazione didattica: cosa significa davvero QFE	6

*Dipartimento Costruzioni Aeronautiche – ITIS “Ettore Majorana”
Via Sant’Angelo, 2 – 03043 Cassino (FR)
rauucci@gmail.com

5	Accelerazione verticale netta: dalla misura in g ai m/s²	6
5.1	Il problema: l'accelerometro misura anche la gravità	6
5.2	Perché è didatticamente importante	6
6	Procedura operativa: connessione, calibrazione, pausa	7
6.1	Calibrazione	7
6.2	Pausa della telemetria	7
6.3	Loop di aggiornamento a 20 Hz	7
7	Interfaccia grafica e interpretazione dei grafici	8
8	Limiti del modello e sviluppi futuri	8
9	Conclusioni	9

1 Introduzione e obiettivi didattici

Nella strumentazione aeronautica reale, la quota non è semplicemente “un numero” da leggere su un display: è una grandezza che dipende dal modello atmosferico, dal riferimento scelto (QNH/QFE/QNE) e dalla qualità del sensore. Allo stesso modo, l’accelerazione verticale è un’informazione essenziale per interpretare le variazioni di assetto e di moto, ma richiede sempre un passaggio concettuale: distinguere la parte dovuta al moto dalla parte dovuta alla gravità.

L’obiettivo di questo progetto è costruire un sistema di telemetria *semplice e trasparente*, pensato per studenti, in cui ogni blocco software corrisponde a un’idea fisica precisa:

- la **quota barometrica** è ricavata dalla pressione misurata dal sensore;
- l’**accelerazione verticale netta** è ottenuta sottraendo la gravità misurata durante una fase di calibrazione;
- i dati sono acquisiti via BLE e resi visibili in tempo reale su due grafici con assi etichettati e unità;
- l’operatore può **mettere in pausa** la telemetria senza alterare la coerenza temporale dei grafici.

La scelta di non introdurre subito algoritmi di fusione complessi (come il Filtro di Kalman) è deliberata: per studenti alle prime armi è più importante comprendere bene i mattoni fondamentali (sensori, riferimento, calibrazione, unità, tempi) prima di introdurre uno stimatore avanzato.

2 Setup sperimentale e architettura del sistema

2.1 Nodo sensore: TI CC2650 SensorTag

Il nodo di acquisizione è un **Texas Instruments CC2650 SensorTag**, un kit IoT compatto che integra diversi sensori e comunica via Bluetooth Low Energy (BLE). Nel progetto vengono utilizzati in particolare:

- il canale barometrico (pressione statica);
- il canale di movimento (accelerometro), da cui estraiamo la componente lungo asse z .



Figura 1: Esempio di SensorTag CC2650 utilizzato come nodo di telemetria.

2.2 Stazione di controllo a terra

Il programma Python realizza una *Ground Station* didattica:

- connessione BLE e sottoscrizione alle notifiche dei sensori;
- interfaccia grafica con pulsanti operativi (CONNETTI, CALIBRA, PAUSA);
- due grafici temporali sincronizzati:
 1. quota (m) vs tempo (s);
 2. accelerazione verticale netta (m/s^2) vs tempo (s).

3 Comunicazione BLE e acquisizione dati

3.1 Principio di funzionamento BLE: notifiche GATT

Il protocollo BLE, in questo caso, viene utilizzato in modalità tipica IoT:

- il sensore espone dei servizi e delle caratteristiche (*UUID*);
- il PC si connette al dispositivo e attiva la trasmissione;
- i dati arrivano come **notifiche** asincrone (*callback*).

Nel codice, l’attivazione avviene in modo chiaro:

Listing 1: Attivazione sensori e sottoscrizione alle notifiche BLE

```

1 await self.client.write_gatt_char(PRESS_CONFIG_UUID, bytearray([0x01])
  )
2 await self.client.write_gatt_char(MOV_PERIOD_UUID, bytearray([10]))
3 await self.client.start_notify(PRESS_DATA_UUID, self.on_press)
4 await self.client.start_notify(MOV_DATA_UUID, self.on_mov)

```

3.2 Decodifica della pressione

La pressione viene ricostruita dalla porzione di payload BLE:

Listing 2: Decodifica pressione dal pacchetto BLE

```
1 p = int.from_bytes(data[3:6], byteorder='little') / 100.0
2 self.raw_press = p
```

A livello didattico, è importante sottolineare che:

- il sensore trasmette un pacchetto binario;
- il software lo interpreta come numero intero e lo scala;
- il risultato `raw_press` deve essere in unità coerenti con la formula di quota utilizzata (vedi section 4).

3.3 Decodifica dell'accelerazione lungo z

Dal pacchetto movimento si estrae la componente z :

Listing 3: Decodifica accelerazione lungo Z

```
1 val = int.from_bytes(data[10:12], byteorder='little', signed=True)
2 self.raw_accel_z = val * (8.0 / 32768.0)
```

Il fattore $\frac{8}{32768}$ è una scala tipica quando il range è impostato a $\pm 8g$. Il risultato `raw_accel_z` è quindi espresso in “g” (cioè multipli dell'accelerazione di gravità).

4 Altimetria barometrica in modalità QFE

4.1 Idea fisica: pressione come “altezza”

Nella troposfera, la pressione decresce con la quota. Un altimetro barometrico misura la pressione statica e, assumendo un modello atmosferico, la converte in quota.

In questo progetto lavoriamo in modalità **QFE**:

- P_0 è la pressione al suolo misurata in fase di calibrazione;
- la quota calcolata vale $h = 0$ al suolo (al momento della calibrazione).

4.2 Formula utilizzata

La relazione implementata nel codice è:

$$h_{\text{baro}} = 44330 \left[1 - \left(\frac{P}{P_0} \right)^{\frac{1}{5.255}} \right], \quad (1)$$

dove:

- P è la pressione istantanea misurata;
- P_0 è la pressione di riferimento a terra (QFE).

Nel codice:

Listing 4: Quota barometrica QFE calcolata in realtime

```
1 h_baro = 44330.0 * (1.0 - math.pow((self.raw_press / self.p_ground_ref
    ), (1.0/5.255)))
```

4.3 Osservazione didattica: cosa significa davvero QFE

Il valore calcolato non è una quota “assoluta” sul livello del mare, ma una quota *relativa al punto di calibrazione*. Questo approccio è estremamente utile in laboratorio perché:

- elimina la necessità di conoscere QNH o di introdurre correzioni meteo;
- rende immediata l’interpretazione: a terra l’indicazione deve essere circa zero.

5 Accelerazione verticale netta: dalla misura in g ai m/s^2

5.1 Il problema: l’accelerometro misura anche la gravità

Un accelerometro non misura direttamente “la variazione di quota”. Misura l’accelerazione specifica lungo i suoi assi. Quando il drone è fermo a terra, l’accelerometro legge tipicamente circa $1g$ lungo l’asse verticale (a seconda della convenzione di segno e dell’orientamento).

Per ottenere una grandezza più leggibile e utile in telemetria, il codice costruisce l’**accelerazione netta verticale** sottraendo il valore medio misurato durante la calibrazione:

$$a_{z,net} = (a_{z,meas} - a_{z,0})g. \quad (2)$$

dove:

- $a_{z,meas}$ è la misura istantanea (in g);
- $a_{z,0}$ è la media a drone fermo (in g);
- $g \approx 9.81 \text{ m/s}^2$.

Nel codice:

Listing 5: Accelerazione verticale netta in m/s^2

```
1 acc_z_net = (self.raw_accel_z - self.gravity_offset) * 9.81
```

5.2 Perché è didatticamente importante

Questa sottrazione è un passaggio concettuale fondamentale:

- mostra che la misura di un sensore grezzo non è sempre direttamente interpretabile;
- introduce l’idea di **bias** e di **offset** (anche senza statistica avanzata);
- prepara il terreno a concetti futuri: integrazione inerziale, fusione sensoriale, stimatori.

6 Procedura operativa: connessione, calibrazione, pausa

6.1 Calibrazione

La calibrazione è il cuore operativo del sistema, perché definisce i riferimenti:

- pressione di suolo P_0 (QFE);
- offset gravità per l'accelerometro.

Nel codice:

Listing 6: Calibrazione: QFE e offset gravità

```
1 self.p_ground_ref = statistics.median(self.p_buffer)
2 self.gravity_offset = statistics.mean(self.a_buffer)
```

Si usa la mediana per la pressione per ridurre l'effetto di campioni anomali, e la media per l'accelerazione per ottenere una stima stabile del valore "a riposo".

6.2 Pausa della telemetria

La pausa è pensata per attività di laboratorio:

- si può fermare la raccolta dati e commentare la situazione;
- si può riprendere senza falsare l'asse temporale.

Il codice gestisce la pausa tenendo traccia del tempo totale in pausa, sottratto al tempo visualizzato:

$$t_{\text{plot}} = (t - t_0) - T_{\text{pausa}}. \quad (3)$$

Nel codice:

Listing 7: Tempo "pulito" per il grafico (pause escluse)

```
1 elapsed = time.time() - self.start_time - self.total_paused_duration
```

6.3 Loop di aggiornamento a 20 Hz

L'interfaccia non si aggiorna quando arrivano i pacchetti BLE, ma con un timer regolare (50 ms):

- garantisce fluidità grafica;
- separa acquisizione asincrona e visualizzazione sincrona;
- evita blocchi della GUI.

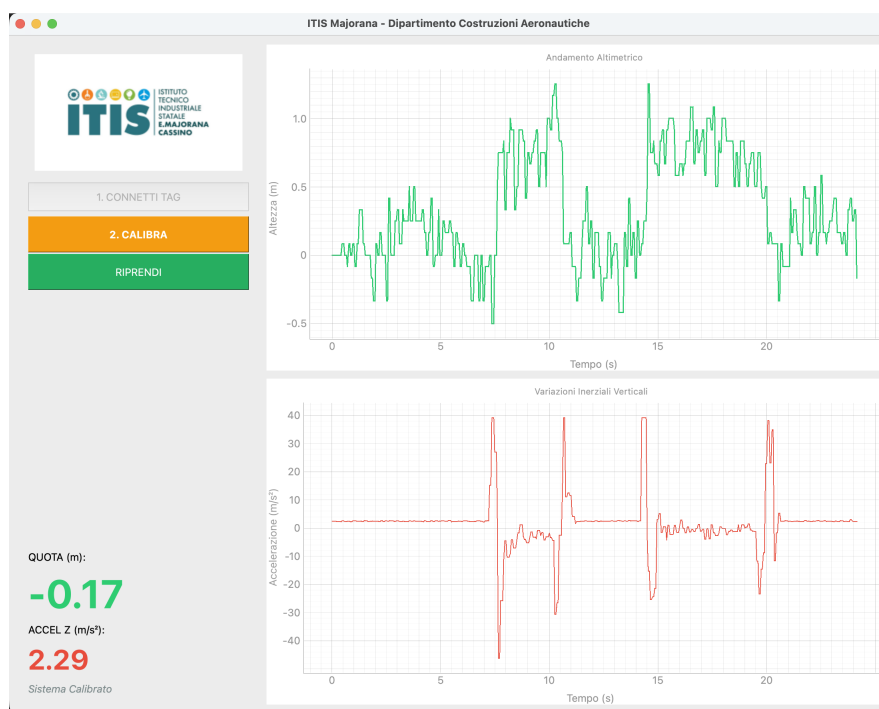


Figura 2: Esempio di dashboard: grafico quota (m) e accelerazione verticale netta (m/s^2).

7 Interfaccia grafica e interpretazione dei grafici

La dashboard mostra due grafici sincronizzati sul tempo:

1. **Andamento altimetrico:** utile per osservare decollo, salita, hovering e discesa.
2. **Variazioni inerziali verticali:** utile per osservare accelerazioni impulsive (es. variazione di thrust, turbolenze, vibrazioni).

È importante sottolineare che:

- la quota barometrica può risultare rumorosa (turbolenza sul sensore, vibrazioni, disturbi di pressione);
- l’accelerazione netta può mostrare spike dovuti a vibrazioni strutturali o manovre rapide.

8 Limiti del modello e sviluppi futuri

Il sistema presentato è volutamente essenziale: misura e visualizza due grandezze fondamentali, ma non stima direttamente la velocità verticale né applica fusione sensoriale avanzata.

I principali limiti (utili anche come spunti didattici) sono:

- la quota barometrica è sensibile a disturbi locali di pressione e non fornisce dinamica molto rapida;

- l'accelerazione non viene integrata per stimare quota o velocità (evitando volutamente problemi di deriva);
- non è presente un filtro di fusione (es. Kalman), che rappresenta uno sviluppo naturale del progetto.

Sviluppi futuri coerenti con questo lavoro possono essere:

- introduzione di un filtro passa-basso sull'accelerazione netta;
- stima della velocità verticale tramite integrazione con correzione barometrica;
- implementazione di un Filtro di Kalman 1D come evoluzione didattica (seconda fase del percorso).

9 Conclusioni

Il progetto realizza una telemetria didattica completa e robusta, fondata su concetti aeronautici essenziali:

- definizione del riferimento altimetrico (QFE) tramite calibrazione;
- lettura e conversione dei dati sensoriali con unità fisiche coerenti;
- distinzione tra misura grezza e grandezza fisica interpretata (accelerazione netta);
- visualizzazione in tempo reale con gestione corretta del tempo e delle pause.

La semplicità del sistema, unita alla trasparenza del codice, lo rende particolarmente adatto a lezioni di laboratorio, verifiche pratiche e introduzione alla strumentazione aeronautica e ai sistemi di acquisizione dati.