Alma Mater Studiorum · Università di Bologna
Scuola di Ingegneria e Architettura · Sede di Forlì

CORSO DI LAUREA IN INGEGNERIA AEROSPAZIALE
Classe L-9

ELABORATO FINALE DI LAUREA
In Meccanica del Volo

Sviluppo e sperimentazione di sensori per il monitoraggio della qualitá dell'aria

Candidato:
Massimiliano Menghini

Relatore:
Prof. Fabrizio Giulietti

Anno accademico 2015/2016
III Sessione

Indice

Elenco delle figure

Elenco delle tabelle

Introduzione

Durante la sua breve presenza sulla terra l'uomo ha sempre utilizzato le risorse a sua disposizione in maniera indiscriminata, senza considerare gli effetti che le sue scelte avrebbero avuto sull'ambiente: un tempo, infatti, la popolazione umana era decisamente meno numerosa, e come conseguenza le azioni degli uomini registrarono un impatto pressoché limitato di danni all'ambiente. Oggi, invece, a causa dell'enorme crescita demografica legata allo sviluppo dei paesi emergenti e all'incremento del tasso di sviluppo nei paesi già ricchi, le azioni degli uomini generano giorno dopo giorno, sia al livello locale che globale, disastri ambientali senza precedenti con ricadute sempre più gravi per l'uomo e per l'intera vita sulla terra.
Per questo motivo, negli ultimi decenni, la questione ha toccato sempre più gli interessi delle grandi potenze, tanto da aprire un dibattito ormai molto caldo e inarrestabile. Infatti, risulta effettivamente molto difficile individuare un'unica soluzione, esclusiva ed efficiente tale da tutelare gli interessi che stanno a cuore ad ognuno, seppur i tentativi non siano mancati. Uno di questi è rappresentato dal protocollo di Kyoto, redatto nel 1997, ed entrato in vigore solo nel 2005, il cui obiettivo si prefiggeva di ridurre le emissioni di gas serra entro il 2012. L'innovazione più grande apportata dal trattato, inoltre, fu quella di operare una distinzione tra i paesi ricchi e i paesi poveri del mondo, motivo per cui all'epoca Cina ed India risultarono escluse dal regime di limitazioni imposte sulle sostanze inquinanti. Vennero, dunque, riconosciute le responsabilità dei maggiori produttori di CO_2, a fronte di oltre 150 anni di attività industriale, in base alle quali ricadde un onere maggiore per le nazioni sviluppate. Nonostante la larga adesione al protocollo, il suo successo si è rivelato comunque più che limitato, per via delle ratifiche non avvenute, come nel caso degli Stati Uniti, e dell'etichetta di paesi poveri attribuita ai casi già menzionati come quelli di Cina e India, che sono divenuti intanto tra i maggiori produttori al mondo di gas serra.
Individuare e conoscere la natura degli inquinanti atmosferici e disporre dei dati delle emissioni sono azioni fondamentali per formulare politiche ambientali incentrate sul miglioramento della qualità dell'aria, e monitorarne

l'efficacia. Sorge l'esigenza di un controllo costante della qualità dell'aria, processo che avviene utilizzando delle centraline di monitoraggio fisse sparse nelle vie delle maggiori città o nei pressi dei principali insediamenti industriali. Il monitoraggio occasionale o periodico di un sito, con strumenti portatili, installati su un veicolo o portati a tracolla da un operatore, è anch'esso un'esigenza in tutti quegli ambiti dove non sia giustificato, soprattutto dal punto di vista economico, tenere operativa una rete fissa.

Lo scopo di questo progetto è quello di realizzare una stazione di monitoraggio mobile al fine di aumentare la superficie di controllo, realizzando un oggetto dinamico capace di acquisire dati sull'inquinamento. Questo è stato fatto applicando ad un drone un sistema di sensori capaci di rilevare le variazioni dei livelli di concentrazione degli agenti inquinanti. Ciò permette di eliminare le stazioni di monitoraggio fisse, le quali rappresentano una spesa ingente. Inoltre, attraverso l'utilizzo di un drone, è possibile monitorare siti più vasti, permettendo un monitoraggio costante e ripetuto nel tempo.

L'elaborato è suddiviso in tre parti. Il Capitolo 1 analizza il sistema *Embedded* utilizzato per l'acquisizione dei dati, concentrando l'attenzione prevalentemente sui moduli utilizzati.
In seguito, il Capitolo 2 descrive quali sono i primi passi per cominciare ad utilizzare i sensori posti sulla *Gases Board 2.0*. Inoltre sarà trovata risposta ai dubbi più comuni, come: quali parametri di configurazione adottare per un avere una risposta adeguata; quale processo di calibrazione seguire; come trasformare i dati acquisiti, espressi in tensioni, in valori di concentrazione di gas.
Il Capitolo 3, infine, illustra i test effettuati per verificare il corretto funzionamento del sistema completo, con l'esposizione delle problematiche individuate, e una presentazione delle alternative più valide per superarle.

1 Sistema Embedded per l'acquisizione dati

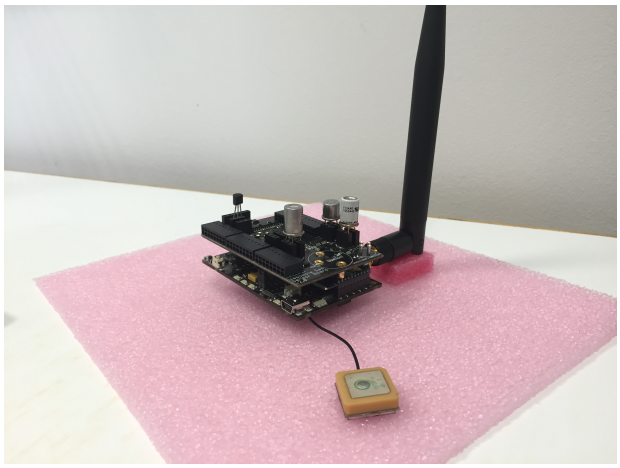
Affinché la veridicità di ogni esperimento sia garantita, è necessario andare ad analizzare non solo i dati raccolti durante le fasi di sperimentazione, ma è altrettanto importante indagare sia le componenti della strumentazione utilizzata per eseguire i test, che la programmazione che sta alla base di tutti questi esperimenti. Per questo motivo risulta utile iniziare il lettore ad una conoscenza più approfondita degli elementi che costituiscono la base fondamentale di questa sperimentazione. Pertanto introduciamo la board che costituisce lo strumento fondamentale sul quale è stato sviluppato tutto il progetto, ovvero il *Waspmote* dell'azienda *Libelium*. Il *Waspmote* è sviluppato su un'architettura di tipo modulare, che permette l'integrazione di moduli adatti esclusivamente ad una determinata missione. La scheda è stata progettata sul modello *Arduino*. Lo scopo dei progettisti era quello di realizzare una scheda adatta per qualsiasi tipo di utilizzo, che fosse di piccole dimensioni, leggera, con consumi piuttosto contenuti. In particolare, grazie alle sue caratteristiche e funzionalità, il sistema permette di effettuare le seguenti azioni:

- Misurare la concentrazione delle particelle di specifici gas;
- Possibilità di sostituire il set di sensori con altri per gas di altro tipo, o per misurare altre grandezze;
- Possibilità di sincronizzare i dati acquisiti con dati di posizione;
- Possibilità di memorizzare i dati nella SD presente nella *board*;
- Effettuare la telemetria con una postazione di terra per visualizzare i valori delle misure in tempo reale.

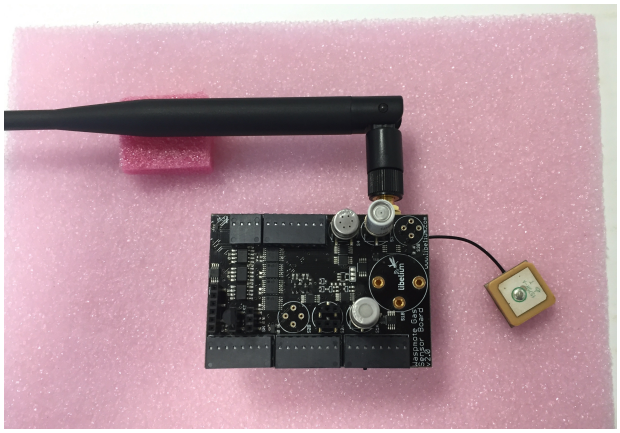
Grazie alla sua versatilità è stato possibile posizionare la board su un mezzo in movimento, quale è un drone, in modo da poter mappare il territorio con le varie concentrazioni di gas presenti nell'ambiente.

I moduli disponibili in Waspmote sono suddivisi in :

- Main Board;
- Modulo GPS;
- Moduli XBee di bassa e alta potenza, utilizzati per la comunicazione;
- Modulo Sensori (*Gases Board 2.0*);
- Modulo di raccolta dati : utilizzo di una memory card SD (supporto massimo di 2GB);



(a) Sensore



(b) Schema circuito

Figura 1: Waspmote

1.1 Main Board Waspmote

È il modulo principale, essenziale per il funzionamento dell'intero sistema. Sulla board sono presenti un microprocessore ATMEGA 1281, un RTC (Real Time Clock), un Crystal Oscillator, le prese di alientazione (USB, porta batteria), un adattatore per scheda di memoria SD, un accelerometro, e numerose porte di input/output che permettono l'integrazione di altri moduli o sensori.

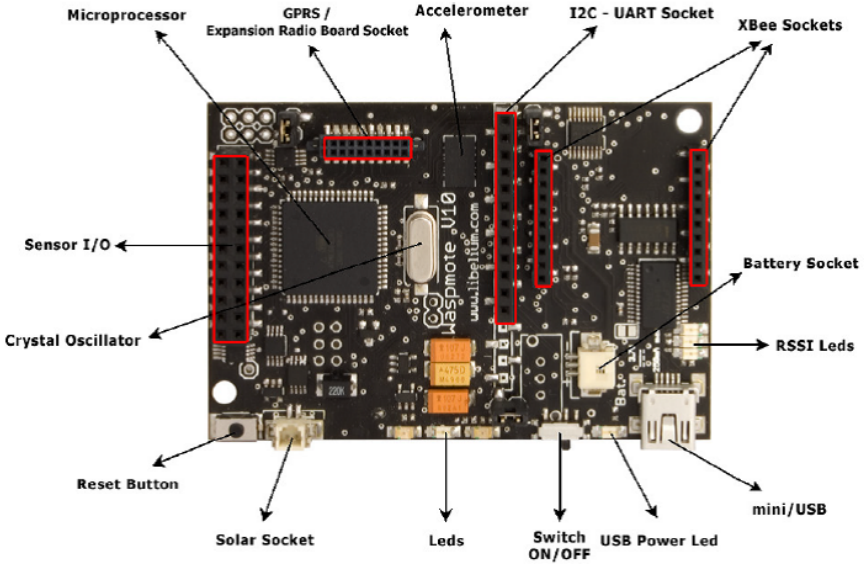


Figura 2: Main Board lato a

Il microprocessore ha una frequenza di 14,7 MHz. Questa unità inizia ad eseguire il bootloader per il caricamento nella memoria dei programmi compilati e delle librerie precedentemente salvate nella memoria Flash, cosicché il programma creato possa essere eseguito. Ogni nuovo programma caricato va a sostituire quello già presente; nel caso non vi sia nessun nuovo programma nel sistema, continuerà a girare l'ultimo caricato. Durante l'esecuzione del codice le variabili dichiarate devono essere salvate in una memoria non volatile, poiché nel caso in cui il dispositivo si spenga, i valori andrebbero persi. Per poter salvare i dati sulla main board è presente un adattatore di memoria SD, che supporta una Sd card di dimensione massimo di 2 GB.

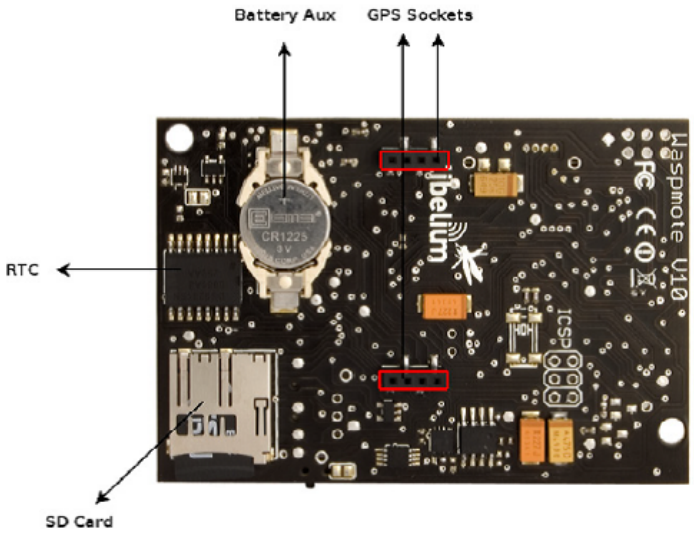


Figura 3: Main Board lato b

Il Real Time Clock è il dispositivo che regola il tempo relativo all'interno del sistema, permettendo di programmare eventuali allarmi per effettuare diverse operazioni in diversi momenti del giorno. Inoltre, permette alla Wasmote di funzionare al massimo del risparmio energetico, e di attivarsi nel momento prestabilito. La board permette di operare in 4 modalità per minimizzare il consumo della batteria:

- **ON**: modalità operativa normale;
- **Sleep**: ogni modulo ha delle funzionalità sospese;
- **Hibernate**: tutte le funzionalità dei moduli sono sospese;
- **OFF**: sistema completamente spento.

1.2 Modulo GPS

Su Waspmote possiamo integrare un ricevitore GPS che ci permette di conoscere la posizione della board in qualsiasi momento. Inoltre è possibile sincronizzare il Real Time Clock con la data e l'ora ricevute dai satelliti.

Il GPS ci fornisce informazioni riguardanti: latitudine, longitudine, quota, velocità, data, ora e efemeridi. Per la ricezione dei dati vengono utilizzati due

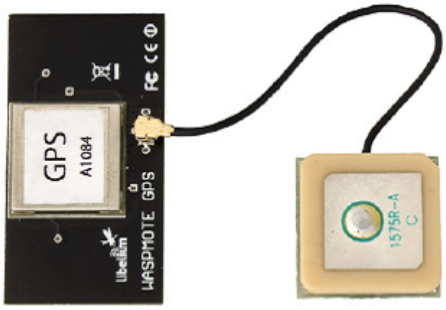


Figura 4: Modulo GPS

diversi tipi di protocollo: NMEA (National Marine Electronic Association) e mdalità binaria. La prima modalità utilizza istruzioni di questo standard per ottenere informazioni su posizione, data e ora. La modalità binaria, invece, permette lo scambio di informazioni tra il ricevitore GPS e il microcontrollore, utilizzando statement specifici di questo protocollo. Le più importanti dichiarazioni del protocollo NMEA sono:

- GGA: fornisce un'indicazione di validità della misura effettuata;
- RMC: forniscono posizione, data e ora;
- GSA fornisce informazioni sullo stati dei satelliti al quale il ricevitore GPS è agganciato.

Il ricevitore GPS nescessita di un intervallo di tempo per ottenere ed elaborare le informazioni inviate dai satelliti. Si può ridurre questo tempo se si è a conoscenza di alcune informazioni preliminari che si trovano nelle effemeridi e negli almanacchi. Le informazioni che possiamo ottenere sono la posizione del satellite (dalle effemeridi) e la traiettoria che percorrerà nei giorni a venire dagli almanacchi. Questi ultimi hanno una validità di 2 o 3 mesi, entre le informazioni delle effemeridi hanno una validità dalle 3 alle 5 ore. A seconda delle informazioni di cui dispone il ricevitore, la fase di avvio può essere di tre tipi:

- Hot Start : stabilita ora e data, con le effemeridi e gli almanacchi validi salvati in memoria. Tempo $<$ 1 secondo;

- Warm Start: ora, data, almanacchi validi, effemeridi mancanti. Tempo < 32 secondi;
- Cold Start: mancano tutte le informazioni. Tempo < 35 secondi.

Il tempo all'avvio si riduce drasticamente se si ha a disposizione le effemeridi, per questo motivo *Waspmote* mette a disposizione una serie di funzioni che permettono di memorizzare le effemeridi nella SD, per poi poterle utilizzare durante la missione. Le funzioni relative al GPS sono presenti nella libreria *WaspGPS.h*.

1.3 Modulo di Comunicazione: XBee-Pro

Il modulo utilizzato per la comunicazione tra il Waspmote e il PC è l' XBee-Pro, prodotto dalla DIGI. Questo dispositivo consente il trasferimento di dati via Radio Frequenza , sfruttando una normale comunicazione seriale. Ciò significa che per riuscire a trasferire dei dati, senza fili, è necessario utilizzare la periferica UART presente nel microprocessore e la porta serial del PC. Il modulo utilizza una frequenza di 2,4 GHz, suddivisa in 16 canali da 5 MHz ognuno, riuscendo ad avere un raggio d'azione fino a 7 chilometri. Utilizza il protocollo di comunicazione 802.15.4., cioè uno standard per le comunicazioni wireless definito dall' IEEE. L'IEEE è un istituto di ingegneria che si occupa di definire gli standard di comunicazione in modo da garantire l'interoperabilità anche tra moduli realizzati da diversi produttori.

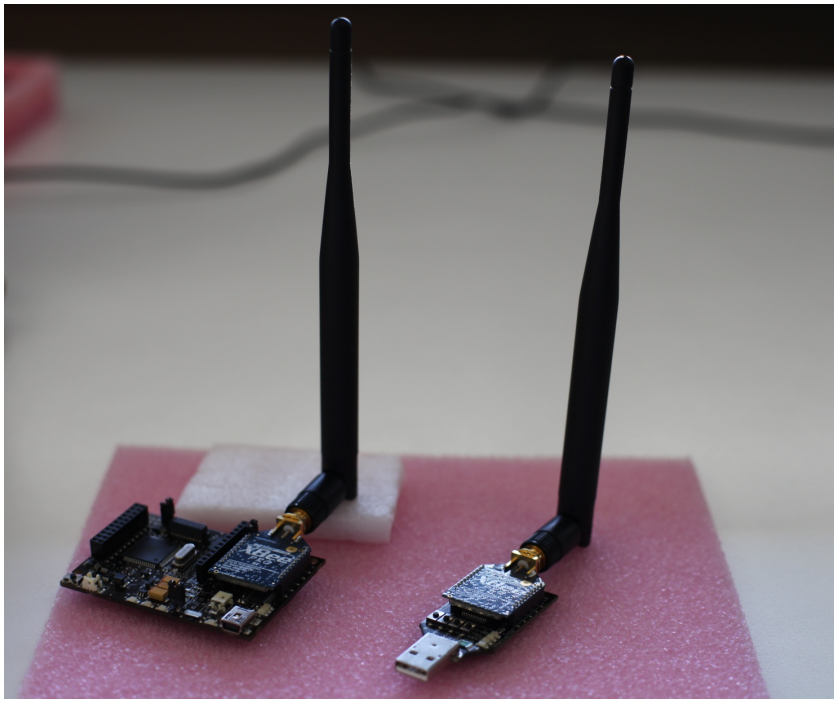


Figura 5: Modulo XBee-Pro

Lo standard 802.15.4 soddisfa i seguenti requisiti:

- Basse velocità di trasferimento;
- Connettività semplice;
- Possibilità di operae con batterie.

Le frequenze definite da questo standard operano nella banda ISM (Industrial,Scientific an Medical), cioè nello spettro elettromagnetico riservato alle applicazioni di radiocomunicazione per uso industriale, scientifico e medico. La banda di frequenza utilizzata è approvata in tutto il mondo, ecco perchè i moduli operanti a tale frequenza sono quelli che hanno una maggiore diffusione. La libreria API che contiene i comandi per utilizzare il modulo radio è *WaspXBee802.h*.

1.4 Gases Board 2.0

Questa è l'unità fondamentale sul quale si è sviluppato tutto il lavoro. La *board* è stata progettata per monitorare parametri come la temperatura, l'umidità, la pressione atmosferica e 14 tipi di gas. Può supportare fino a 6 sensori contemporaneamente. Attraverso l'ambiente di sviluppo IDE è possibile impostare l'amplificazione del segnale di output di ogni sensore, fino ad un guadagnno massimo di 101.

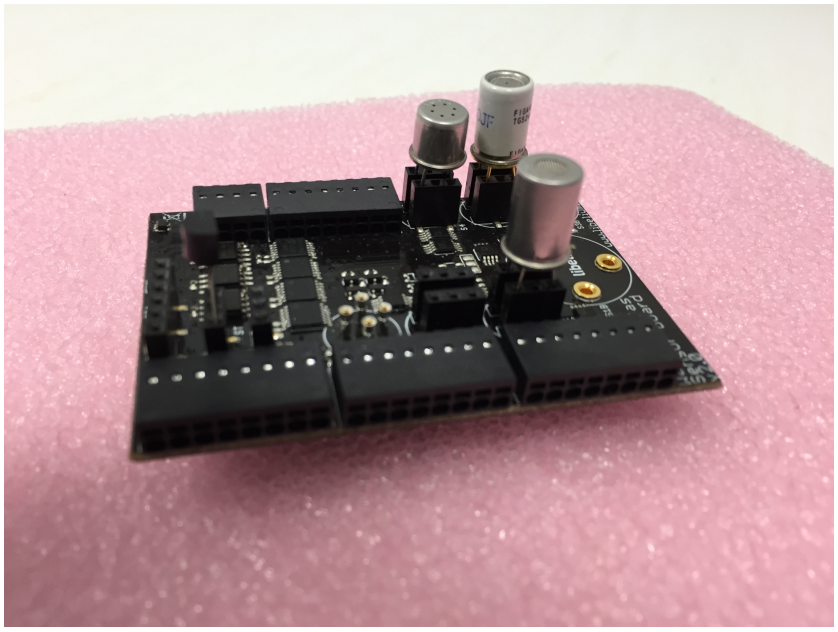


Figura 6: Gases Board 2.0

I gas che possono essere monitorati sono i seguenti:

- Monossido di carbonio - CO;
- Biossido di carbonio - CO_2;
- Ossigeno molecolare - O_2;
- Metano - CH_4;
- Idrogeno molecolare - H_2;
- Ammoniaca - NH_3;

- Isobutano - CH_4;
- Etanolo - CH_3CH_2OH;
- Toluene - $C_6H_5CH_3$;
- Acido Solfidrico - H_2S;
- Diossido di Nitro - NO_2;
- Ozono - O_3;
- Composti Organici Volatili (VOC);
- Idrocarburi.

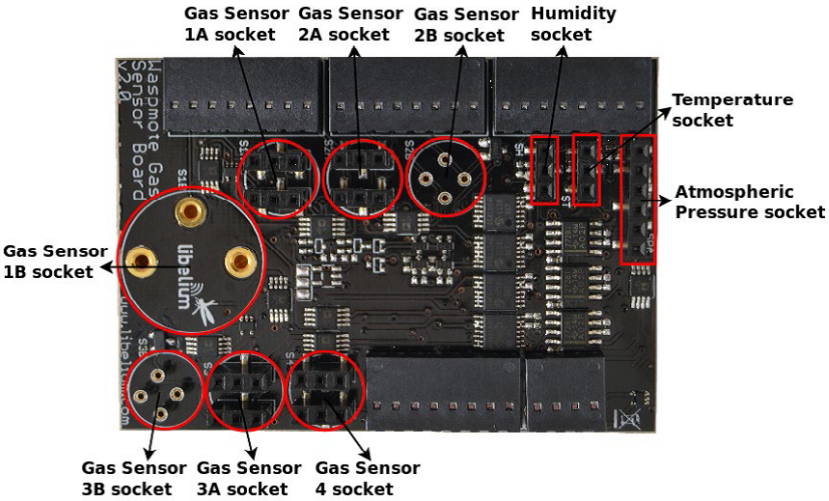


Figura 7: Socket Gases Board

Ogni sensore ha una sua specifica presa. Un carico di resistenza è posto all'uscita di ogni sensore , eccetto per i connetori 1A e 1B in cui non è necessario, combinato con una fase di amplificazione con guadagno massimo di 101. La scelta dei parametri di guadagno e del carico di resistenza si effettua tenendo conto dello specifico sensore che si ha a disposizione , in quanto potrebbero esserci delle diversità anche tra due sensori dello stesso tipo. La misura in uscita dei sensori è espressa in Volt, per cui sarà necessaria una conversione

pee poter anallizzare i dati. Il guadagno e i carico di resistenza possono essere configurati utilizzando i comandi presenti nella libreria *SensorGasv20*. I sensori utilizzati per il monitoraggio della qualità dell'aria sono i seguenti:

- TGS 4161 - CO_2 (Socket 1A);
- TGS 2442 - CO (Socket 3A);
- TGS 2602 - Contaminanti dell'aria (Socket 4A);
- MCP9700A- Temperatura (ANALOG1).

1.5 Programmazione

Per poter comunicare con la board è necessario utilizzare l'apposito ambiente di programmazione *Waspmote IDE* (Integrated Development Environment). Queso ambiente è stato sviluppato a partire dalla piattaforma open source di Arduino, seguendo lo stesso stile nelle librerie e nelle operazioni. È importante però utilizzare la versione per Waspmote e non quella per Arduino, poichè essa è stata strutturata affinché possa funzionare al meglio. I programmi vengono compilati utilizzando il C++ come linguaggio di programmazione. Il codice generalmente è strutturato in due parti: **Setup** e **Loop**.

Figura 8: IDE con esempio di codice.

Nella Fase di **Setup** generalmente avviene l'inizializzazione dei moduli che verranno utlilizzati durante la missione, con le relative variabili. Il processore eseguirà le azioni in maniera sequenziale. Il **Loop** invece, è la parte di programma che ciclicamente si ripete. Questa fase ha una durata infinita a meno che non venga inserita un'interruzione. Al termine del Loop è consuetudine porre la funzione *delay(time)*, la quale rappresenta l'intervallo di tempo che il dispositivo dovrà attendere prima di riprendere un nuovo ciclo. All'interno delle parentesi si inserisce il tempo espresso in millisecondi. Le librerie API sono fornite dalla Libelium e contengono le funzioni relarive ad ogni sensore/modulo che verrà utilizzato con il Waspmote 1.1. Un modo per visualizzare una determinata funzione è quella di consultare il file *".h"* relativo al dispositivo. In questo elenco di funzioni vengono indicati i tipi di input e output, con la relativa descrizione della funzione.

2 Sensori

Quando si lavora con la maggior parte dei sensori utilizzati sulla scheda *Gases Board 2.0*, dal momento che la loro normale resistenza e sensibilità potrebbe variare da un'unità all'altra in un'ampia gamma, è necessario calibrare il sensore ottenendo un valore preciso. Il processo di calibrazione richiede di catturare la risposta del sensore sotto differenti concentrazioni di gas nel *"target range"* operativo, teoricamente compreso nel campo di funzionamento del sensore. A seconda delle condizioni di applicazione da realizzare, è necessario correggere i valori in base alla temperatura e all'umidità, poiché modificherebbero il valore di output. Ovviamente maggiore è il numero di punti di calibrazione nel range di sensibilità, maggiore sarà l'accuratezza di calibrazione. Tuttavia, bisogna considerare che i vari sensori sono sensibili alla variazione di concentrazione di più gas: i dati acquisiti devono essere rielaborati e interpretati utilizzando il relativo datasheet del sensore per poter fare una stima delle concentrazioni dei gas misurati.

Questo procedimento potrebbe rivelarsi superfluo nel caso in cui le misurazioni effettuate siano ripetibili senza dover utilizzare uno specifico equipaggiamento. Se, infatti, l'esperimento è ripetibile non è necessario calibrare il sensore, evitando di dover collezionare i dati a diverse concentrazioni dello stesso gas, e successivamente interpolarli per ottenere la curva di calibrazione. Sarà sufficiente impostare il valore di *offset* a determinate condizioni, ed in seguito servirsi delle equazioni ottenute dai datasheet, ricavando il valore dei liveli di concentrazione.

2.1 Sensore per il rilevamento di CO_2 - TGS4161

Il sensore TGS 4161 è un sensore di tipo solido elettrolitico per il rilevamento di CO_2, il quale fornisce una tensione di output proporzionale alla concentrazione della CO_2 presente nell'atmosfera. Per una concentrazione di circa 350 ppm, che è il normale livello di CO_2 contenuto nell'aria, il sensore mostra dei valori di tensione compresi tra 220 e 440 mV, che decrescono all'aumentare della concentrazione di gas. Al fine di ottenere misurazioni affidabili è necessario calibrare il componente prima dell'utilizzo. L'intervallo di misurazione del sensore è molto ampio, da 350 a 10000 ppm, il che lo rende ideale per il controllo della qualità dell'aria sia in ambiente interno che esterno.

Gas rilevati	CO_2
Range di misura	$350 \sim 10000$ ppm
Tensione a 350 ppm	$220 \sim 490$ mV
Temperatura operativa	$-10 \sim +50$ °C
Tempo di risposta	1,5 minuti
Consumo medio	50 mA

Tabella 1: Specifiche *TGS 4162*

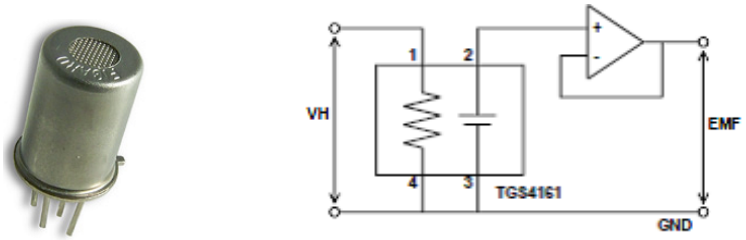
L'elemento sensibile è costituito da un solido elettrolitico posizionato tra due elettrodi. Affinché possa raggiungere la temperatura adeguata alla rilevazione di gas, vi è un heater composto da diossido di rutenio (RuO_2), utilizzato per la sua elevata capacità termica. Monitorando la differenza di forza elettromotrice (ΔEMF) generata tra i due elettrodi, è possibile misurare la concentrazione di gas. La parte superiore del sensore, che ne costituisce il tappo, contiene al suo interno un elemento assorbente, ovvero la zeolite: questa ha come obiettivo quello di ridurre l'influenza di interferenza dei gas. Il sensore *TGS4161* richiede una tensione di ingresso di riscaldamento (V_H), applicata all'heater integrato. La forza elettromotrice (EMF) del sensore è misurata da un amplificatore operazionale ad alta impedenza con una bassa corrente di polarizzazione. Dato che il sensore funziona come una batteria, il valore di forza elettromotrice si ottiene misurando la tensione ai capi dei due poli. Inoltre, il dispositivo presenta una relazione lineare tra la variazione di EMF e la concentrazione di CO_2 su scala logaritmica, mostrando una buona

stabilità dovuta a variazioni di temperatura e umidità. L'accuratezza offerta da questo sensore potrebbe variare a seconda del tempo di alimentazione prima della misurazione. In questo caso un tempo di 30 secondi è sufficiente per cogliere le variazioni di concentrazione, mentre se è richiesta una misurazione più accurata, saranno necessari almeno 10 minuti. Per poter accedere al valore di output, è necessario posizionare il sensore nel connettore 1A ed utilizzare le funzioni fornite nella libreria *SensorGasv20* (Figura 9).

```
{
  SensorGasv20.setBoardMode(SENS_ON);
  SensorGasv20.configureSensor(SENS_CO2, GAIN);
  SensorGasv20.setSensorMode(SENS_ON, SENS_CO2);
  delay(TIME);
  value = SensorGasv20.readValue(SENS_CO2);
}
```

Figura 9: Esempio script per TGS 4161

La funzione *setBoardMode*, che serve per alimentare la *Gases board*, deve essere richiamata prima di utilizzare qualsiasi sensore. Per impostare il fattore di amplificazione (GAIN) alla relativa porta (SENS_CO2) si utilizza la funzione *configureSensor*. Successivamente si richiama la funzione *setSensorMode*, la quale alimenta il sensore. Il comando *delay(TIME)* rappresenta l'intervallo di tempo (TIME) espresso in millisecondi, che il dispositivo dovrà attendere prima di procedere con le successive istruzioni. In questo script è, invece, utilizzato affinché l'elemento sensibile possa raggiungere la temperatura ottimale. Infine si ricorre a *readValue* per la lettura del valore di *output*.



(a) Sensore (b) Schema circuito

Figura 10: TGS 4161

2.2 Sensore per il rilevamento di CO - TGS 2442

Il TGS 2442 è un sensore resistivo sensibile alle variazioni di concentrazione di monossido di carbonio (CO) ed in minima parte all'idrogeno (H_2). Riesce a misurare concentrazioni di CO molto elevate, 30 -1000 ppm, che difficilmente possono essere registrate in condizioni ambientali. In presenza di monossido di carbonio la conducibilità elettrica aumenta in funzione della concentrazione del gas presente nell'aria. Il sensore ha una struttura multistrato composta da:

- Elemento sensibile;
- Isolante elettrico;
- Isolante termico;
- Elemento riscaldatore o *heater*.

L'elemento sensibile composto da diossido di stagno viene isolato elettricamente e termicamente, tramite l'impiego in ordine di un film di allumina e uno di vetro. L'heater posto sotto i due isolanti è composto da uno strato di ossido di rutenio (RuO_2), il quale presenta capacità termiche ideali per riscaldare l'elemento sensibile, e portarlo a temperatura ottimale per la rilevazione del gas. Per misurare la resistenza del sensore sull'isolante elettrico vi sono due elettrodi in oro (Au). Il coperchio interno è riempito con un carbone attivo allo scopo di ridurre l'influenza dei gas di rumore.

Gas rilevati	CO
Range di misura	30 $\sim$ 1000 ppm
Resistenza a 100 ppm di CO	13,3 $\sim$ 133 kΩ
Temperatura operativa	-10°C $\sim$ +50 °C
Tempo di risposta	1 secondo
Resistenza di carico minima (R_L)	10 kΩ
Consumo medio	3 mA

Tabella 2: Specifiche *TGS 2442*

Al sensore vengono applicate due tensioni: tensione del circuito (V_C) e tensione di riscaldamento (V_H). La prima è applicata alla resistenza (R_S)

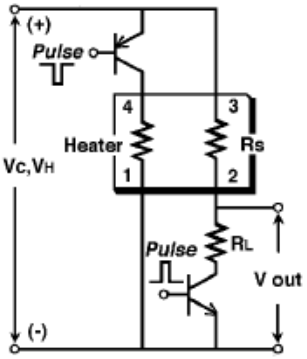
posizionata sull'elemento sensibile e collegata in serie ad una resistenza di carico (R_L), attraverso i due elettrodi del sensore (pin N.2 e N.3). La tensione di riscaldamento, invece, è applicata all'heater, il quale è collegato ai pin N.1 e N.4. La lettura di questo sensore richiede un ciclo di un secondo, durante il quale vengono applicati due impulsi: uno sulla resistenza dell'elemento sensibile (R_S) e l'altra sulla resistenza dell'heater. Il consumo medio del ciclo di alimentazione è di circa 3mA. Per poter utilizzare il sensore è necessario calibrarlo, trovando il valore di resistenza in presenza di 100 ppm di CO, il quale è compreso nel range 13.3 a 133 KΩ. Per attivare il sensore

```
{
  SensorGasv20.setBoardMode(SENS_ON);
  SensorGasv20.configureSensor(SENSOR, GAIN, RESISTOR);
  value = SensorGasv20.readValue(SENSOR);
}
```

Figura 11: Esempio script per TGS 2442

e leggere l'output verranno utilizzati i comandi presenti nella libreria *SensorGasv20* (Figura 11). Inoltre è importante sottolineare che per il corretto funzionamento del sensore bisogna impostare, utilizzando la funzione *configureSensor*, l'adeguato carico di resistenza (RESISTOR) e il giusto guadagno di amplificazione (GAIN).

(a) Sensore



(b) Schema circuito

Figura 12: TGS 2442

2.3 Sensore per il rilevamento dei contaminanti dell'aria - TGS 2602

Il TGS 2602 è un sensore che reagisce alla presenza di gas inquinanti, variando la sua resistenza. Ha un'alta sensibilità sia a basse concentrazioni di gas maleodoranti, come l'ammoniaca (NH_3) generata da materiali di scarto industriali; sia a basse concentrazioni di composti organici volatili (VOC), come il toluene, ad esempio, utilizzato come diluente nella produzione delle vernici. In assenza di contaminanti nell'aria il sensore presenta una resistenza compresa tra i 10 e i 100 kΩ, a causa di questa variabilità è necessaria una calibrazione prima di utilizzarlo. L'elemento sensibile è costituito da uno strato semiconduttore di ossido metallico, isolato elettricamente da un film di allumina. Il tutto è posizionato al di sopra dell'heater composto da ossido di rutenio (RuO_2), come visto precedentemente per il sensore di CO-TGS 2441. La resistenza del sensore diminuisce all'aumentare della concentrazione dei gas presenti.

Gas rilevati	$C_6H_5CH_3$, H_2S, CH_3CH_2OH, NH_3, H_2
Range di misura	$1 \sim 30$ ppm
Resistenza in aria fresca	$10 \sim 100$ kΩ
Temperatura operativa	$+10$°C $\sim +50$ °C
Tempo di risposta	30 secondi
Resistenza di carico minima (R_L)	0,45 kΩ
Consumo medio	61 mA

Tabella 3: Specifiche *TGS 2602*

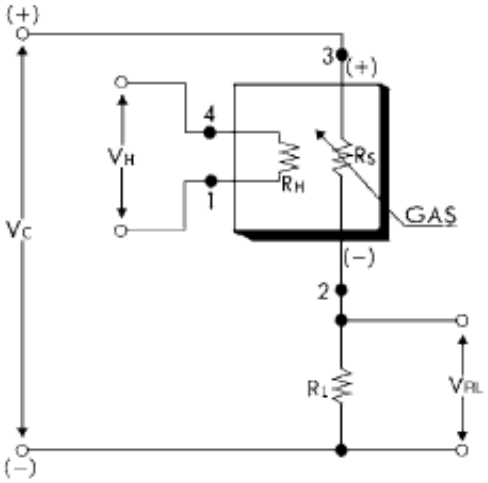
Il sensore richiede due ingressi di tensione: una tensione di riscaldamento (V_H) e una tensione del circuito (V_C). La tensione di riscaldamento viene applicata all'heater, per mantenere l'elemento sensibile ad una temperatura ideale per il rilevamento dei gas, mentre la seconda viene applicata alla resistenza di carico per consentire la misurazione della tensione (V_{out}), necessaria per poter ricavare i livelli di concentrazione. Per permettere all'elemento sensibile di raggiungere la temperatura adeguata è necessario aspettare un tempo di 30 secondi prima di effettuare qualsiasi misurazione.

```
{
  SensorGasv20.setBoardMode(SENS_ON);
  SensorGasv20.configureSensor(SENSOR, GAIN, RESISTOR);
  SensorGasv20.setSensorMode(SENS_ON, SENSOR);
  delay(TIME);
  value = SensorGasv20.readValue(SENSOR);
}
```

Figura 13: Esempio script per TGS 2602

Per poter essere utilizzato, il sensore deve essere posizionarlo nel connettore 2A, facendo riferimento alle funzioni della libreria *SensorGasv20*. Nella Figura 16 è riportato un'esempio di codice per l'acquisizione dei dati. Le funzioni richiamate sono le stesse dei sensori precedenti.Si sottolinea l'importanza di impostare il giusto tempo di riscaldamento prima di utilizzare la funzione *readValue*, per leggere la misura del sensore.

(a) Sensore



(b) Schema circuito

Figura 14: TGS 2602

2.4 Sensore di temperatura - MCP9700A

Il sensore MCP9700A è un sensore analogico che converte il valore della temperatura in un proporzionale valore analogico di tensione. Nonostante sia un sensore economico, a bassa potenza, si ha una precisione di ±2 gradi da 0 a 70°C. A differenza dei sensori resistivi, per questo non c'è bisogno di integrare un circuito addizionale per riscaldare l'elemento sensibile, composto da un diodo. Inoltre, a differenza degli altri sensori non necessita né di una fase di amplificazione, né di una resistenza di carico. Per questo sensore non è necessaria una calibrazione, in quanto risulta già tarato per il range di temperatura, nel quale ci appresteremo ad effettuare gli esperimenti (0 - 70 °C). La Libelium ha implementato nel codice di programmazione la funzione che

Misure rilevate	Temperatura
Range di misura	-40°C $\sim$ +125°C
Precisione	10 ±2°C tra 0°C $\sim$ +70°C
Tempo di risposta	1,65 secondi
Consumo tipico	6 μA
Consumo medio	12 μA

Tabella 4: Specifiche *MCP9700A*

trasforma i valori di tensione in valori di temperatura, di conseguenza non c'è bisogno di dover ricavare alcuna funzione dal datasheet per poter interpretare le tensioni di uscita. Essendo un sensore analogico deve essere posizionato nella porta ANALOG1. L'output del sensore pertanto può essere direttamente letto dal Waspmote tramite la funzione readValue, appartenente alla libreria *sensorGasv20*, che ne cattura il valore analogico e automaticamente lo trasforma in valore di temperatura.

Figura 15: MCP9700A

2.5 Calibrazione

Il metodo che segue può essere applicato a qualsiasi sensore fornito dalla Libelium, fatta eccezione per il sensore di CO_2, il quale ha un processo di calibrazione particolare che verrà presentato nel corso dell'elaborato. Prima di procedere con la calibrazione dei sensori, bisogna tenere conto delle seguenti considerazioni:

- A causa della bassa precisione e ripetibilità, ogni sensore deve essere calibrato e dovrà avere i propri parametri di calibrazione;
- La calibrazione migliora il comportamento del sensore, ma non garantisce una risposta ad alta precisione;
- Il processo di calibrazione deve essere effettuato in un laboratorio con ambiente di prova controllato;
- L'aspettativa di vita dei sensori di gas è di qualche mese, dopo questo periodo i sensori devono essere sostituiti e calibrati;
- La sensibilità del sensore potrebbe cambiare quando il dispositivo è soggetto a grandi variazioni di temperatura o umidità;
- Il tempo di stabilizzazione dei sensori è di circa 10-20 minuti.

Il primo passo sarà quello di conoscere i parametri di configurazione di ogni sensore prima di proseguire con la calibrazione stessa.

2.5.1 Gain

Il guadagno di amplificazione serve per avere una risoluzione maggiore del valore di output. Quando non si amplifica (GAIN = 1), il valore di uscita sarà compreso tra 0 e ciò che il sensore rileva in un ambiante normale. Tuttavia, ci potrebbero essere delle piccole variazioni che non possono essere notate poiché il range di output è troppo piccolo. Aumentando il fattore di amplificazione si ottengono limiti superiori maggiori, cioè si avranno output di uscita compresi tra 0 e 3.3 V riuscendo così a rilevare anche piccole variazioni di concentrazione. Bisogna fare attenzione quando si imposta il Gain, in

quanto si deve evitare di portare a saturazione il componente, cioè ottenendo valori di tensione in uscita superiori a 3.3V, limite superiore dell'hardware. Il Gain può essere configurato tramite il software con una valore minimo di 1 e massimo di 101. Generalmente il Gain sarà fissato a 1 per la maggior parte delle applicazioni, solo in specifiche situazioni, come il rilevamento del gas nei limiti del campo di sensibilità del sensore, sarà necessario scegliere un valore differente.

2.5.2 Resistenza di carico R_L e Resistenza del sensore R_S

Il sensore di rilevazione di CO e quello dei contaminanti dell'aria hanno circuiti simili, con un partitore di tensione. Il partitore di tensione è composto da due resistenze: resistenza del sensore R_S e resistenza di carico R_L. Il valore della prima varia in relazione alla concentrazione di gas presente, mentre la seconda è configurata dalla funzione *configureSensor*. La scelta del valore di R_L cambia in base al tipo di sensore utilizzato, come mostrato nella Tabella 5.

Sensore	**R_L**
CO - TGS 2442	10 kΩ fino a 100 kΩ
TGS 2620 - VOC	0,45 kΩ fino a 100 kΩ
CO2 - TGS 2461	Non necessario
Tempeatura	Non necessario

Tabella 5: Range di R_L

Per il sensore di temperatura e per il sensore di CO_2 non è necessario impostare un carico di resistenza, poichè la misura e il processo di calibrazione sono differenti. Ad esempio nel caso del sensore di CO_2 per misurare la concentrazione di gas si utilizza la differenza di forza elettromotrice compresa tra il valore V_0, definito in condizioni standard, e il valore letto in uscita. La misura in uscita dai sensori è espressa in Volt, per cui sarà necessario convertirla in resistenza equivalente, utilizzando la relazione 1.

$$R_S = \left(\frac{V_C \cdot R_L}{V_{out}}\right) - R_L \tag{1}$$

- R_S resistenza del sensore;
- V_C tensione di alimentazione della *board* 5V;
- V_{out} tensione misurata dal sensore;
- R_L resistenza di carico impostata dall'utente.

2.5.3 Resistenza R_0

Il valore R_0 del sensore corrisponde alla resistenza misurata in condizioni standard, ovvero ad una temperatura di 25 °C, in condizioni di umidità normali (40% RH), con una determinata concentrazione di gas nota o in aria fresca. Tale valore verrà utilizzato per convertire i valori in parti per milione. La Tabella 6 mostra in che condizioni deve essere misurato R_0 in base al tipo di sensore utilizzato.

Sensore	**R_0**
CO - TGS 2442	resistenza R_S a 100 ppm di CO
TGS 2620 - VOC	resistenza R_S in aria fresca
CO2 - TGS 2461	resistenza R_S in aria fresca
Tempeatura	Non necessaria

Tabella 6: Condizioni per misurare R_0

2.5.4 Fase di Calibrazione

Questo processo può essere applicato a qualunque sensore fornito dalla Libelium fatta eccezione per i sensori di temperatura e di CO_2, poichè il primo ha una risposta lineare e non è necessaria la sua calibrazione, mentre il secondo utilizza normalemente una tensione invece di una resistenza.

1. Configurare il sensore con la funzione *configureSensor()*. Il guadagno di amplificazione (GAIN) è normalemente impostato ad 1, mentre la resistenza di carico R_L può variare a seconda del sensore.

```
{
#define GAIN 1 //guadagno di amplificazione
#define RL 50 // resistenza di carico in KOHM
//congigurazione sensore
SensorGasv20.configureSensor(SENS_SOCKET4A,GAIN,RL);
}
```

2. Attendere il tempo necessario affinchè il sensore raggiunga la temperatura ideale per la misurazione del gas, richiamando la funzione *setSensorMode* e *delay()*.

```
{
  #define TIME 30000 //tempo espresso in millisecondi
  //accensione sensore
  SensorGasv20.setSensorMode(SENS_ON,SENS_SOCKET4A);
  delay(TIME)
}
```

3. Leggere il valore V_{out} ad una concentrazione nota, controllando le condizioni di temperatura ed umidità.

```
}
 //lettura valore
 value = SensorGasv20.readValue(SENS_SOCKET4A);
{
```

4. Trasformare i valori letti in resistenza utilizzando la relazione 1.

```
{
  resistance = calcola_res(RL,value);

  //funzione che calola la resistenza dei sensori
  float calcola_res (float load_sens, float tensione)
  {
    float resistenza;
    resistenza = (5*load_sens)/tensione - load_sens;
    return (resistenza);
  }
```

5. Trovare il valore R_0 come valor medio di tutti i dati acquisiti durante la misurazione.

2.5.5 Calibrazione per sensore CO_2 - TGS 4261

Il sensore *TGS 4161* fornisce una tensione di output proporzionale alla concentrazione di CO_2 presente nell'atmosfera. Per poter utilizzare questo sensore è necessario impostare il valore di offset in condizioni standard, cioè ad una concentrazione di 350 ppm di CO_2, il normale livello presente in ambiente esterno. A questa concentrazione il dispositivo mostra valori compresi tra 220 mV e 490 mV. Per poter impostare il valore corretto, data la grande variabilità che il sensore prensenta a questa concentrazione, si sono effettuate delle prove sperimentali che riproducessero le condizioni standard. Per ottenere un range di output più esteso, il guadagno di amplificazione (Gain) è stato impostato al valore 7, che ne ha aumentato la risoluzione. Sono stati effettuati 6 esperimenti in giorni diversi con umidità diverse, posizionando il sensore all'esterno del laboratorio di Meccanica del Volo. I test prevedevano una durata di 20 minuti, affinchè la risposta del dispositivo potesse stabilizzarsi. Al termine di ogni prova è stata calcolata la tensione di offset come il valor medio di tutti i dati acquisiti. Il valore finale al contrario è stato calcolato facendo la media dei valor medi di tutti i test effettuati. Lo script utilizzato è lo stesso della Figura 9.

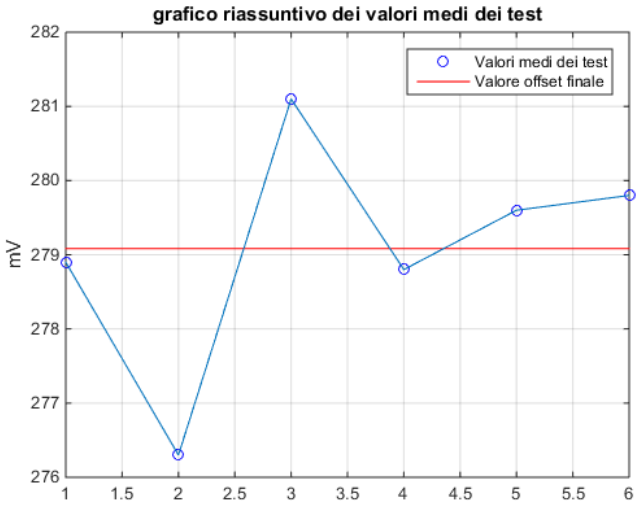


Figura 16: Valori medi dei test

Test	Valore Medio [mV]	Temperatura	Umidità
1°	278,9	8 °C	90%
2°	276,3	10 °C	85%
3°	281,1	9 °C	87%
4°	278,8	8 °C	88%
5°	279,6	8,5 °C	86%
6°	279,8	9 °C	88%

Tabella 7: Risultati dei test di calibrazione del sensore *TGS4161*

Dai test effetuati risulta un valore medio di 279.03 mV, il quale verrà utilizzato come valore di offset. Il dispositivo risulta insensibile a variazioni di umidità ad una concentrazione di 350 ppm di CO_2. In Figura 17 si osserva come il valore di forza elettromotrice (EMF) rimanga costante al variare del'umidità relativa. Nonostante i test siano stati effettuati con un'umidità relativa superiore all'80%, il valore di offset calcolato è considerato accettabile.

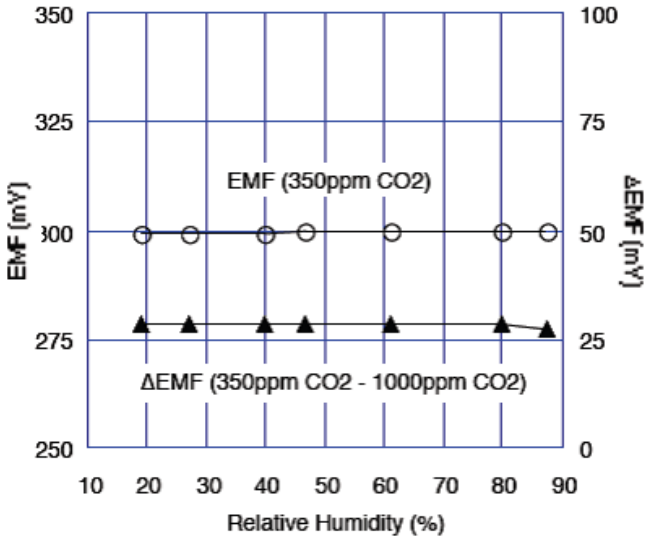


Figura 17: Variazione nei dati in funzione dell'umidità

2.6 Analisi dei dati

I dati acquisiti devono essere rielaborati per far si che venga disposta una lettura più chiara delle misure effettuate. Risulta necessario, quindi, trasformare le tensioni e le resistenze in parti per milione, per ottenere una lettura più chiara delle concentrazioni di gas misurate. Affinché ciò sia possibile, è necessario ricavare le relazioni matematiche dai datasheet di ogni sensore.

2.6.1 Sensore CO_2

Il sensore fornisce come output il valore della forza elettromotrice (EMF) che si genera tra i due elettrodi. Per prima cosa si calcola il ΔEMF, ossia la differenza di potenziale espressa in mV. Qusta differenza non è altro che quella tra la tensione di offset, precedentemente misurata, ad una concentrazione di 350 ppm di CO_2 e la tensione di output del sensore.

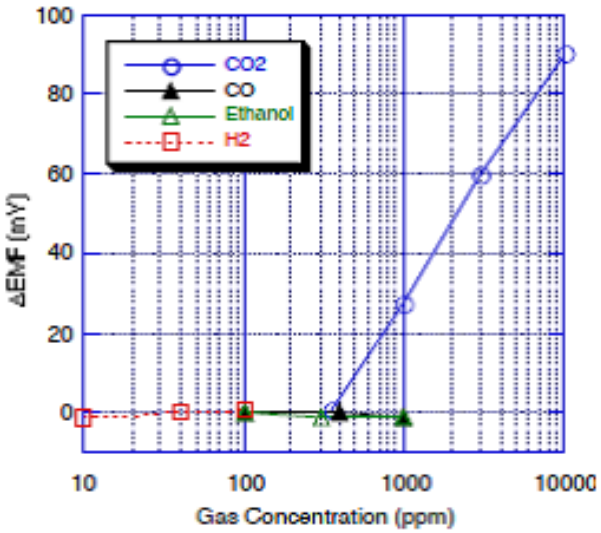


Figura 18: Datasheet *TGS 4161*

Per cui:

$$\Delta EMF = V_{350ppm} - V_{out} \tag{2}$$

La relazione tra la ΔEMF e la concentrazione di gas è di tipo lineare. Partendo dai punti noti del datasheet (Figura 18) è possibile calcolare la relazione

che lega la differenza di tensione alle ppm:

$$\Delta EMF = 64,305 \cdot \log_{10}(ppm) - 163,596 \tag{3}$$

Esplicitando le ppm otteniamo:

$$ppm = 10^{\frac{\Delta EMF + 163,596}{64,305}} \tag{4}$$

2.6.2 Sensore CO

Il TGS 2442 è sensibile a variazioni di concentrazione di CO e di H_2 come mostrato nel datasheet in Figura 19. I dati raccolti devono essere espressi come rapporto tra la resistenza del sensore (R_S) e la resistenza di offset (R_0), ottenuta dalla calibrazione.

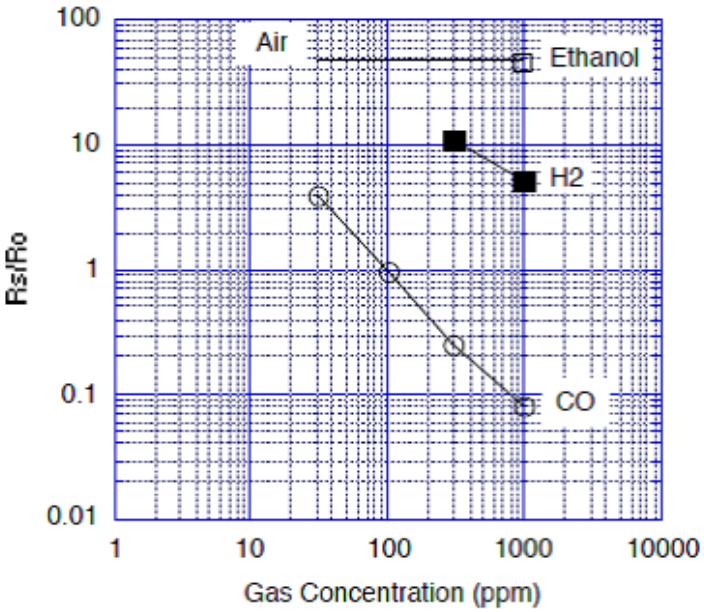


Figura 19: Datasheet *TGS 2441*

In base ai valori di input, asse delle ordinate, si determina la presenza di un tipo di gas. Per poter interpretare i campioni acquisiti, occore dividerli in tre marcro intervalli (Tabella 8) .

Intervallo	**Gas rilevato**
$10 < R_S/R_0 < 40$	Non c'è presenza né di CO, né di H_2
$5 < R_S/R_0 < 10$	Presenza di H_2
$0,08 < R_S/R_0 < 4$	Presenza di CO

Tabella 8: Intervalli di sensibilità del sensore *TGS4221*

Per ottenere le equazioni di trasformazione si ricorre alla seguente equazione:

$$\log_{10}(\frac{R_S}{R_0}) = m \cdot \log_{10}(ppm) + b \tag{5}$$

dove **m** è il coefficiente angolare e **b** il termine noto. Calcolati m e b a partire dai punti noti dal grafico ($\circ$ per la curva CO e $\square$ per la retta H_2) ed esplicitando le ppm si ottiene la relazione di conversione. La curva relativa all'andamento di CO è divisa in tre tratti di rette su scala logaritmica, perciò è necessario utilizzare tre diverse relazioni in base al valore del campione. Le rette sono state ricavate partendo dai punti noti del datasheet. Di seguito sono riportate le equazioni di conversione con i relativi intervalli di validità .

$$se \quad 1 < \frac{R_S}{R_0} < 4 \qquad ppmCO = 10^{\dfrac{-\log_{10}(\frac{R_S}{R_0}) + 2,302}{1,151}} \tag{6}$$

$$se \quad 0,25 < \frac{R_S}{R_0} < 1 \quad ppmCO = 10^{\dfrac{-\log_{10}(\frac{R_S}{R_0}) + 2,520}{1,262}} \tag{7}$$

$$se \quad 0,08 < \frac{R_S}{R_0} < 0,25 \quad ppmCO = 10^{\dfrac{-\log_{10}(\frac{R_S}{R_0}) + 1,741}{0,946}} \tag{8}$$

Invece se il dato è compreso nell'intervallo di presenza di H_2 si utilizza:

$$ppmH_2 = 10^{\dfrac{-\log_{10}(\frac{R_S}{R_0}) + 2,475}{0,592}} \tag{9}$$

2.6.3 Sensore contaminanti dell'aria

Il TGS 2602 è sensibile a diversi tipi di gas comunemente denominati composti organici volatili, come mostrato nel datasheet in Figura 20. Come per il sensore TGS 2442 è necessario adimensionalizzare il dato acquisito dividendolo per il valore R_0.

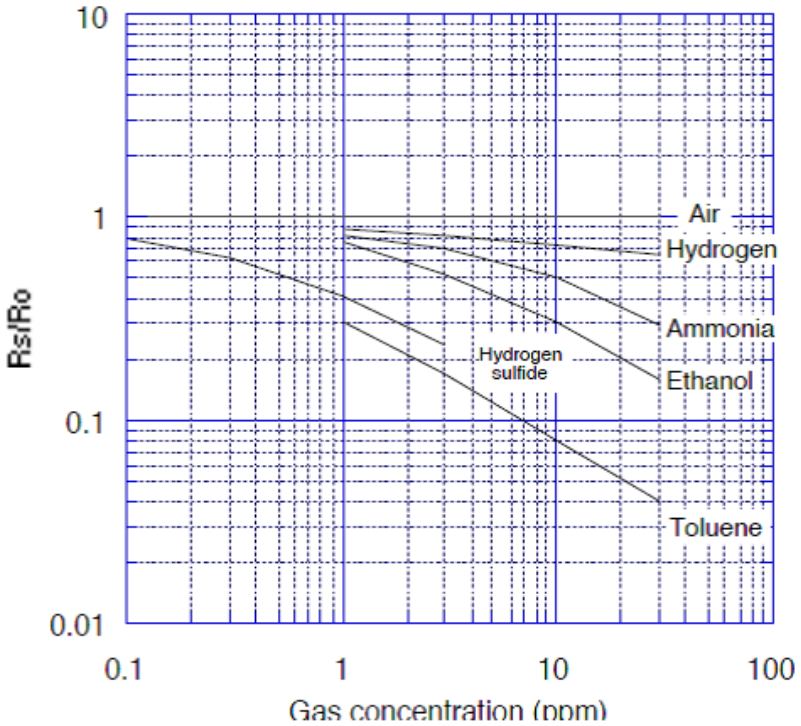


Figura 20: Datasheet *Datasheet TGS 2602*

Dato un valore di input il sensore avverte la presenza di più gas. Ad esempio se si ottiene un valore di $\frac{R_S}{R_0}$ di 0.4, si è in presenza di 1 ppm di acido solfidrico, 6 ppm di Etanolo e 16 ppm di Ammoniaca. Le curve del'andamento dei vari gas sono tutte composte da delle rette a tratti in scala logaritmica. Per ricavare le equazione di conversione si utilizza la relazione 5, prelenvando dei punti noti dal datasheet. di seguito sono riportate le relazioni per ogni tipo di gas al quale il TGS 2602 è sensibilie, con il relativo intervallo di validità.

Toluene :

$$se \quad 0,04 < \frac{R_S}{R_0} < 0,3 \qquad ppm = 10^{\frac{\log_{10}(\frac{R_S}{R_0}) + 0,5229}{-0,5924}} \tag{10}$$

Acido Solfidrico H_2S:

$$se \quad 0,6 < \frac{R_S}{R_0} < 0,8 \qquad ppmH_2S = 10^{\frac{\log_{10}(\frac{R_S}{R_0}) + 0,3588}{-0,2619}} \tag{11}$$

$$se \quad 0,4 < \frac{R_S}{R_0} < 0,6 \qquad ppmH_2S = 10^{\frac{\log_{10}(\frac{R_S}{R_0}) + 0,3979}{-0,3368}} \tag{12}$$

$$se \quad 0,25 < \frac{R_S}{R_0} < 0,4 \qquad ppmH_2S = 10^{\frac{\log_{10}(\frac{R_S}{R_0}) + 0,3979}{-0,6781}} \tag{13}$$

Ammoniaca NH_3:

$$se \quad 0,7 < \frac{R_S}{R_0} < 0,8 \qquad ppmNH_3 = 10^{\frac{\log_{10}(\frac{R_S}{R_0}) + 0,0969}{-0,1215}} \tag{14}$$

$$se \quad 0,5 < \frac{R_S}{R_0} < 0,7 \qquad ppmNH_3 = 10^{\frac{\log_{10}(\frac{R_S}{R_0}) + 0,0216}{-0,2795}} \tag{15}$$

$$se \quad 0,3 < \frac{R_S}{R_0} < 0,5 \qquad ppmNH_3 = 10^{\frac{\log_{10}(\frac{R_S}{R_0}) + 0,1639}{-0,4650}} \tag{16}$$

Etanolo:

$$se \quad 0,5 < \frac{R_S}{R_0} < 0,8 \qquad ppm = 10^{\frac{\log_{10}(\frac{R_S}{R_0}) + 0,0969}{-0,4278}} \tag{17}$$

$$se \quad 0,3 < \frac{R_S}{R_0} < 0,5 \qquad ppm = 10^{\frac{\log_{10}(\frac{R_S}{R_0}) + 0,0986}{-0,4243}} \tag{18}$$

$$se \quad 0,17 < \frac{R_S}{R_0} < 0,3 \qquad ppm = 10^{\frac{\log_{10}(\frac{R_S}{R_0}) + 0,0059}{-0,5170}} \tag{19}$$

Idrogeno Molecolare H_2:

$$se \quad 0,65 < \frac{R_S}{R_0} < 0,9 \qquad ppmH_2 = 10^{\frac{\log_{10}(\frac{R_S}{R_0}) + 0,0458}{-0,0957}} \tag{20}$$

3 Sperimentazione

3.1 Esperimenti Funzionamento dei sensori

Ciò che ci si è proposti di osservare nei due seguenti esperimenti è la verifica del corretto funzionamento dei sensori che rilevano la temperatura ed i livelli di CO_2. Perciò, è stato modificato il sistema nel quale è stata effettuata l'acquisizione dei dati, cercando di evidenziare le loro variazioni durantee i test. Gli esperimenti sono stati effettuati nel laboratorio di meccanica di volo, ed i risultati ottenuti sono stati analizzati ed interpretati mediante l'utilizzo dello script Matlab, in modo tale da ottenere una lettura più semplice dei risultati collezionati. Questo metodo ci ha consentito di convertire misure di voltaggi direttamente in ppm. Per effettuare questa operazione è stato necessario utilizzare le equazioni precedentemente calcolate dai diversi datasheet dei sensori.

3.1.1 Esperimento Temperatura

Il primo esperimento è atto a verificare variazioni nei dati acquisiti dal sensore di temperatura, effettuando delle prove cambiando, tramite *script*, il tempo di acquisizione, al fine di leggere un valore il più vicino possibile a quello reale, misurato con il termometro. Un errato tempo di acquisizione porta ad una errato funzionamento del sensore, poiché i dati collezionati sono affetti da disturbi causati da tempi di ascolto elevati.

Condizioni al Contorno:

- temperatura iniziale misurata con termometro: 22.3°C;
- temperatura finale misurata con il termometro: 23.5°C;
- tempo acquisizione scelto: 5 secondi.

Il test è stato effettuato in laboratorio, riscaldando il sensore di temperatura con un phon, di seguito è riportato il profilo di temperatura:

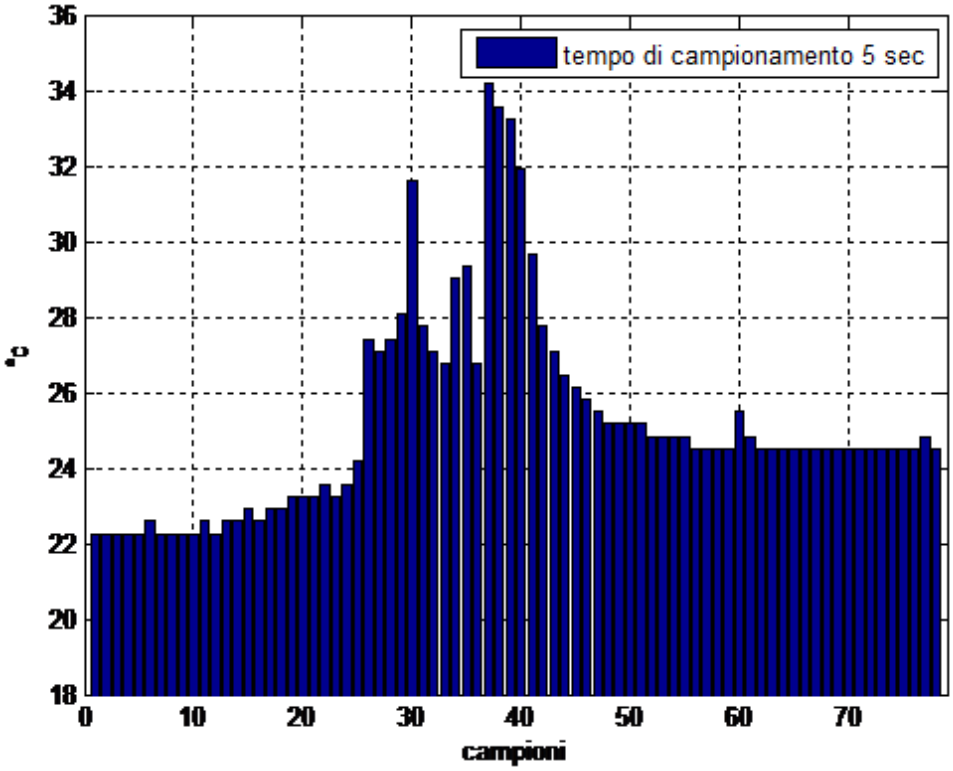


Figura 21: Profilo di Temperatura

La temperatura iniziale registrata dal sensore è stata di 22.25°C, quindi molto vicina alla misura registrata dal termometro. Prima di cominciare a riscaldare l'ambiente abbiamo aspettato affinché il sensore si stabilizzasse. Iniziata la fase di riscaldamento, le variazioni registrate con il sensore sono più lente rispetto a quelle registrate con il termometro. si suppone che questo ritardo sia legato al tempo di acquisizione impostato, che tuttavia non può essere inferiore, perchè se si utilizzasse un intervallo più piccolo il sensore registrerebbe valori disturbati. Dopo il picco massimo registrato abbiamo spento il phon, continuando ad acquisire dati. Come detto i tempi di risposta del sensore sono stati molto più lunghi rispetto a quelli del termometro, inoltre la temperatura finale registrata è maggiore di 1.5°C, il che probabilmente è dovuto al fatto che il test è stato effettuato in aria calma, dove il sensore ha

bisogno di più tempo per smaltire il calore. Il tempo di acquisizione migliore durante le prove effettuate si è rivelato essere di 5 secondi, in quanto è il tempo che disturba meno i dati.

3.1.2 Esperimento variazione livelli di CO_2

Per poter verificare le variazioni dei livelli di concentrazione di CO_2, la *Gases Board* è stata inserita in un ambiente isolato con un termometro e una candela accesa.

Condizioni al Contorno:

- temperatura iniziale: 22.3°C;
- temperatura finale : 26.0°C;
- Gain Sensore CO_2: 7;
- Valore di offset calcolato a 350 ppm di CO_2: 315,66 mV;
- Tempo di Campionamento: 35 secondi

Il Valore di offset per il sensore è stato scelto, previa calibrazione, seguendo la procedura standard fornita dalla *Libelium*, cioè posizionando la *Gases Board* in aria fresca, acquisendo i dati e impostando il valore di offset come il valore medio di tutti i valori. Il tempo di campionamento utilizzato è stato selezionato per permettere all'elemento sensibile di raggiungere la temperatura adeguata per l'acquisizione dei dati, evitando la registrazione di valori disturbati. Il *Gain* impostato permette di registrare tutte le variazioni di Gas, al quale il sensore *TGS4161* è sensibile, evitando di portare il sensore in saturazione.

Di seguito sono riportati i grafici relativi all'analisi dei dati:

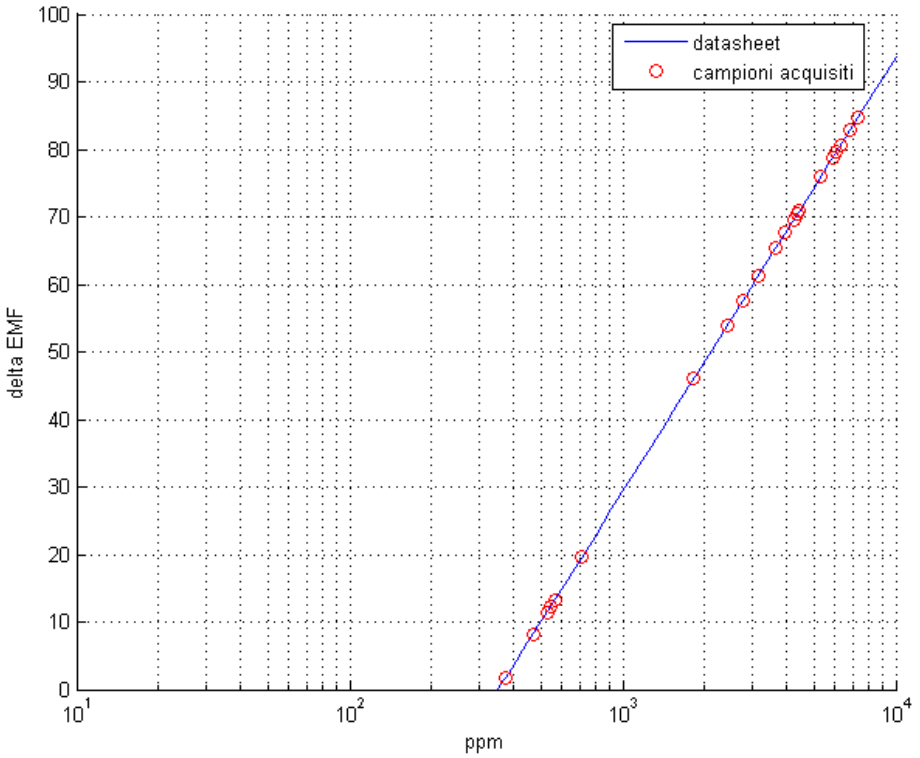


Figura 22: Datasheet *TGS4161* con campioni acquisiti.

In Figura 22 è riportata la curva del datasheet del sensore, in cui vengono evidenziati i campioni acquisiti. Il valore minimo registrato è stato di 445 ppm durante la fase iniziale del test. Il valore massimo, invece, si attesta intorno alle 8800 ppm, picco rilevato nel momento in cuii si è spenta la candela.

Nella figura 23 si può osservare una crescita delle ppm, senza considerare la parte iniziale dove il sensore doveva ancora rilevare la presenza di CO_2. Come già rilevato, il picco massimo è stato registrato nel momento in cui la candela si è spenta. In seguito allo spegnimento della candela, i dati raccolti hanno evidenziato una diminuzione della concentrazione di CO_2 nell'ambiente, presumibilmente correlato al fatto che l'ambiente in cui è stato svolto il test non fosse chiuso ermeticamente.

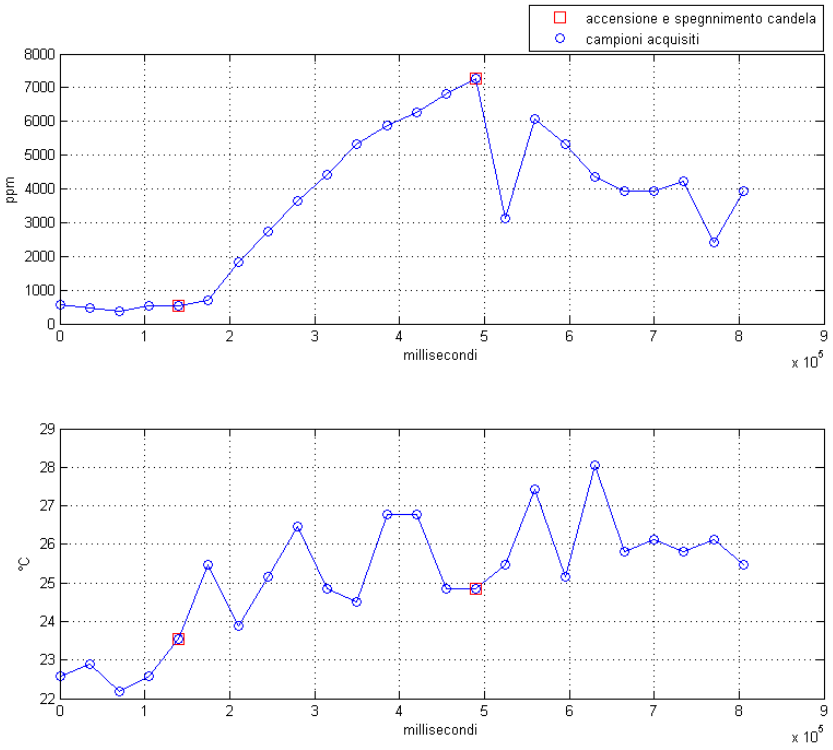


Figura 23: andamento della concentrazione di CO_2 e della temperatura durante il corso dell'esperimento.

Per quanto concerne la variazione della temperatura nel corso del test, è stato notato un leggero incremento, che tuttavia differisce dai valori attesi. Ciò che ci si aspettava era, infatti, un incremento più ripido come si era osservato per il sensore di CO_2. Tuttavia i dati collezionati possonno essere considerati affidabili, sebbene si discostino dai risultati attesi, in quanto non è stato possibile posizionare la candela in prossimità della *board* per motivi di sicurezza. Inoltre le superfici del contenitore non possono essere considerate adiabatiche. L'ambiente di sviluppo della board non permette di acquisire dati in tempi diversi: non è possibile consentire a tutti i sensori di acquisire i dati secondo le loro tempistiche ottimali, perciò è stata privilegiata l'acquisizione dei valori della CO_2, affinché fossero il più precisi possibile. E' evidente che se fosse stato possibile impostare tempi di acquisizione diversi per ogni sensore, il sensore di temperatura avrebbe registrato dati meno disturbati.

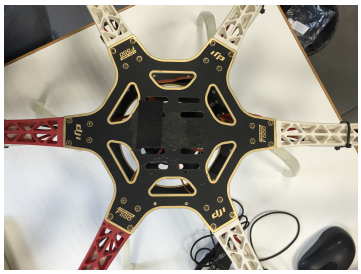
3.2 Disturbi creati dai rotori del S.A.P.R. sull'acquisizione dati

3.2.1 DJI Flame Wheel F550 e PIXHAWK AutoPilot

Nell'ottica di proseguire con la filosofia di progettazione *open source* della *Waspmote*, è scelto di utilizzare come mezzo di movimentazione per i nostri esperimenti il S.A.P.R.,(Sistema Aeromobili a pilotaggio remoto), *F550* della *DJI*, controllato dall'autopilota *PIXHAWK* poichè tutti e due i progetti sono cosiddetti *open*, cioè sviluppati da una *community* di *developers* che rendono accessibile a qualsiasi utente i codici sorgente. Tuttociò ne favorisce il libero studio e permette a qualsiasi utente di apportarvi modifiche ed estensioni.

(a) Motori.

(b) ESC.



(c) Frame

(d) Autopilota

Figura 24: Componenti Flame Wheel F550

Il drone in esame è un multirotore estremamente versatile, adatto sia a piloti esperti, sia a principianti alla ricerca di una facile e divertente ricerca di volo. Il kit fornito dalla *DJI* è composto da:

- *Frame* realizzato con materiale composito ad alta resistenza;

- 6 motori DJI E300 *brushless*: motori elettrici con il rotore a magneti permanenti e lo statore interno a campo magnetico rotante. La contrapposizione dei campi genera la rotazione del rotore a cui è fissata l'elica;
- 6 Electronic Speed Controller DJI 15A: si occupano della gestione della corrente in arrivo al motore e vanno a regolare il numero di giri in base all'input dell'autopilota. Sono essenziali per garantire la variazione di assetto voluta dal pilota e per contrastare gli effetti del vento;
- Carrelo di atterraggio.

3.2.2 Analisi dei disturbi

Dovendo effettuare acquisizione dati in volo in ambiente esterno, è stato necessario verificare se il movimento provocato dai rotori del S.A.P.R. producesse un disturbo sui sensori, influenzando l'acquisizione dei dati ed inficiando i risultati ottenuti. Per sovradimensionare i disturbi è stato poi scelto di utilizzare l'esacottero F550 della DJI, alimentandolo tramite generatore di corrente.

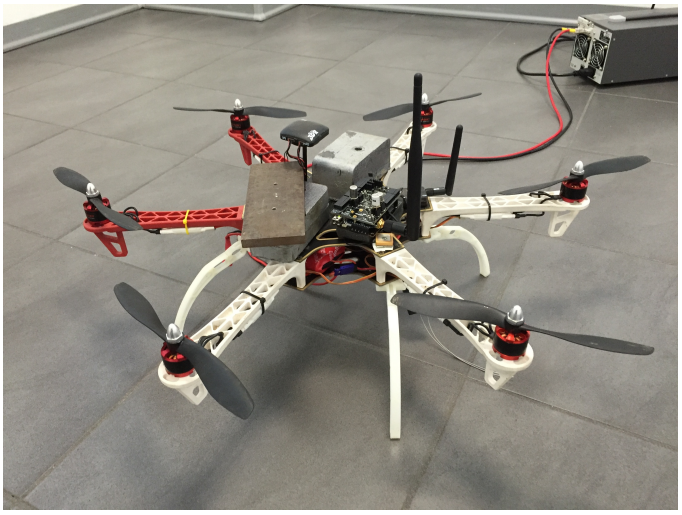


Figura 25: S.A.P.R. F550 con *Waspmote*.

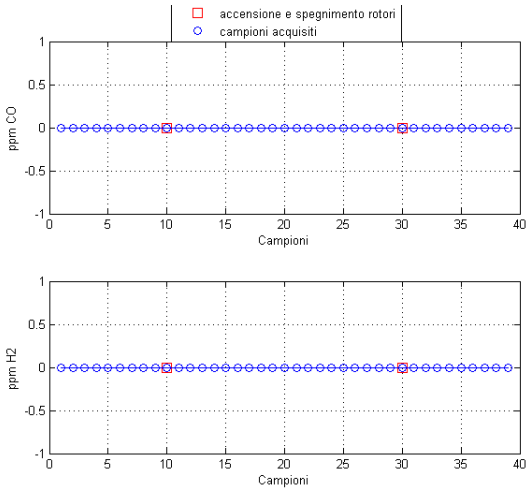
Il primo test è stato effettuato in una stanza vuota del Tecnopolo Aeronautico, posizionando sul frame delle zavorre che permettessero fissare il drone al pavimento. La temperatura della stanza, verificata con un termometro, era di 22.5°C. Il test si è articolato in tre fasi:

1. La *board* è stata attivata a motori spenti, aspettando che i sensori raggiungessero la temperatura ottimale e che le risposte si stabilizzassero;

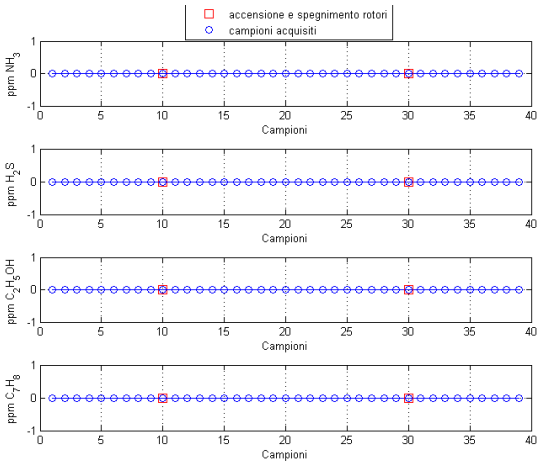
2. Dopo la fase di preriscaldamento dei sensori, il test prevedeva l'accensione dei motori a manetta costante, continuando con l'acquisizione dati.

3. Per verificare se effettivamente il flusso d'aria, generato dalla rotazione delle eliche, inficiasse i valori dei sensori, l'acquisizione dati è continuata a motori spenti.

Di seguito, sono riportati i grafici relativi ai dati acquisiti da tutti i sensori, riportando anche quelli relativi all'andamento del sensore di CO e dei contaminanti dell'aria, sebbene non molto indicativi, poiché non sono state rilevate concentrazioni di gas durante tutto l'arco della prova.



(a) Sensore CO.



(b) Sensore contaminanti dell'aria.

Figura 26: Dati sensori primo Test.

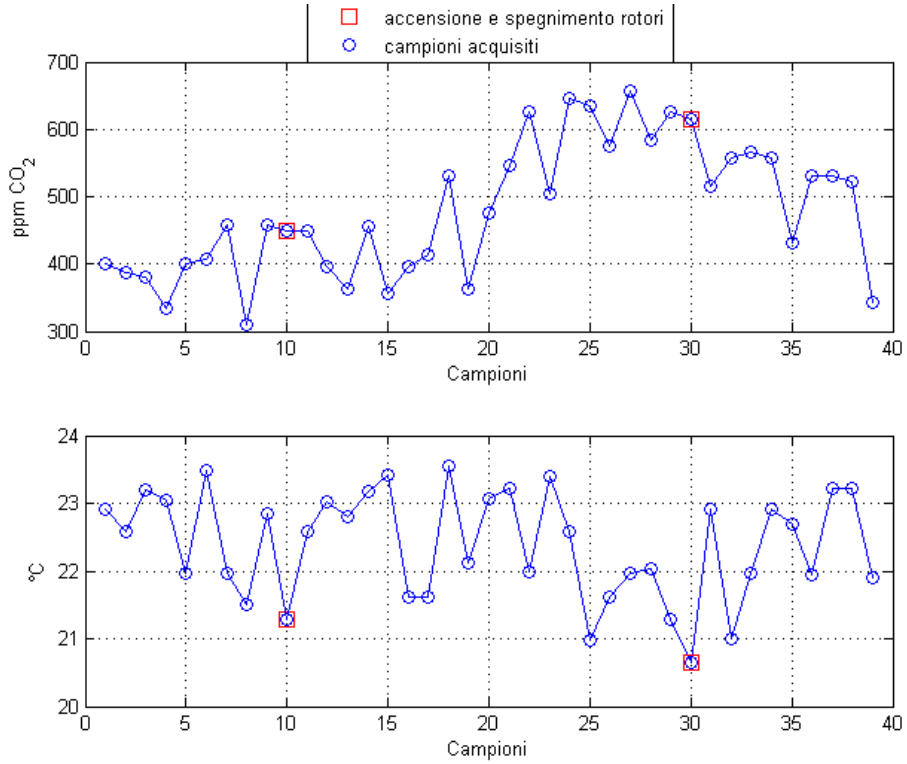


Figura 27: Dati Sensore CO_2 e Temperatura primo Test.

Nel primo grafico della Figura 27 relativo al sensore di CO_2 , durante il funzionamento dei rotori, i dati raccolti dai sensori evidenziano un incremento di concentrazione del gas. Questo aumento di concentrazione non è da attribuire al flusso d'aria maggiore che investe i sensori: infatti non è stato considerato che, essendo rimasti all'interno della stanza durante tutto l'arco del test, i livelli di CO_2 potrebbero essere aumentati involontariamente. Questa ipotesi è riscontrabile nei dati raccolti durante l'ultima parte del test, dove la stanza è stata abbandonata, ma lasciando la porta aperta. Infatti, come si evince dal grafico relativo alla CO_2, c'è un leggera diminuzione di ppm dovuto al ricircolo dell'aria.

Per poter valutare se i sensori fossero effettivamente disturbati dai motori, il test è stato effetuato nuovamente, lasciando isolato tutto il sistema e controllando l'esperimento dall'esterno. I risultati ottenuti sono i seguenti. Non vengno riportati i grafici relativi al sensore di CO e al sensore dei contaminanti dell'aria poichè i sensori non hanno rilevato variazioni.

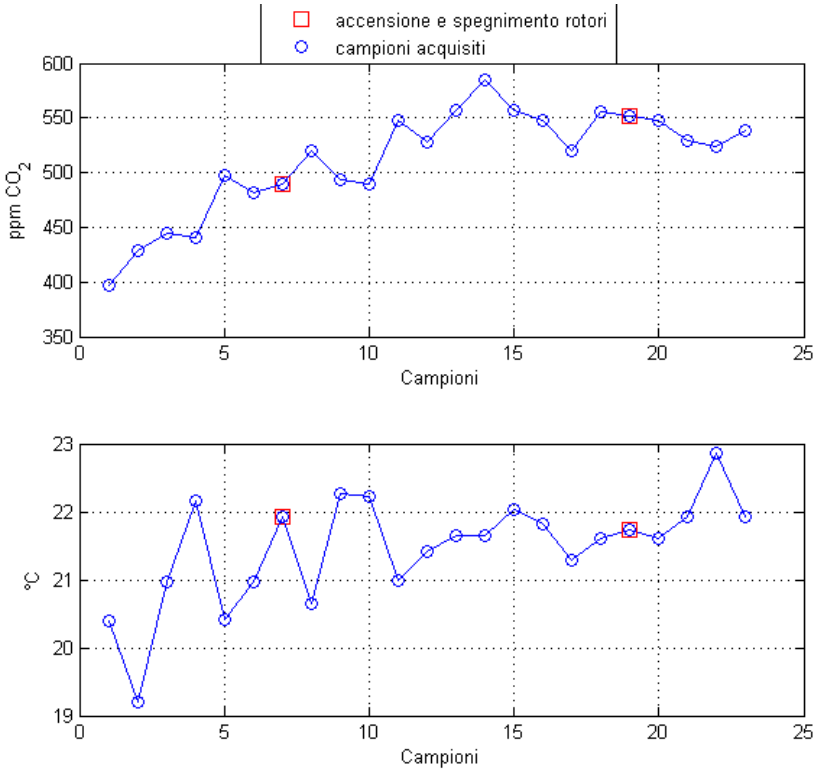


Figura 28: Dati Sensore CO_2 e Temperatura secondo Test.

È evidente che non bisogna considerare i dati acquisiti durante il periodo di preriscaldamento dei sensori, dove i livelli di ppm e di temperatura sono inferiori. Durante la fase di test con motori accesi e in quella successiva le oscillazioni della concentrazione di CO_2 e della temperatura sono minime. Infatti, se nel primo test i dati relativi al CO_2 hanno un'oscillazione di 400 ppm e la temperatura si attesta tra i 21 e i 23 gradi, nel secondo test si ha un'oscillazione inferiore (100 ppm), con la temperatura compresa tra i 21 e 22 gradi. Considerando i valori ottenuti durante i due test, è lecito trarne le seguenti conclusioni: il flusso d'aria creato dalla rotazione delle eliche non influenza i risultati ottenuti, anzi, in un ambiente chiuso come la stanza di test, si innesca un moto convettivo che miscela al meglio tutti i gas presenti nell'ambiente, consentendo di ottenere dei risultati più affidabili.

3.3 Acquisizione dati a terra

L'obiettivo del seguente esperimento è quello di verificare la robustezza del sistema prima di effettuare la prova in volo con il SAPR. È stata simulata un'acquisizione dati con il sistema in movimento al fine di verificare se fossero presenti dei disturbi nell'acquisizione, e per capire al meglio come movimentare il drone in maniera tale da ricevere dei dati GPS più precisi. Dato che il sistema acquisisce sia dati dai sensori che dal GPS ogni 35, è necessario scegliere come far movimentare il drone per minimizzare gli errori in acquisizione. L'esperimento prevede la mappatura dell'esterno del Tecnopolo Aeronautico. La Board è stata programmata in modo da acquisire i valori, salvandoli in file ".txt" all'interno della memory card SD. Inoltre, per poter effettuare un controllo istantaneo sul funzionamento del codice, la board è stata messa in comunicazione con il Pc utilizzando il modulo Radio *XBee Pro*. I dati acquisiti sono stati successivamente elaborati da uno script Matlab, scritto ad-hoc, il quale utilizza le relazioni di conversione già approfondite nel capitolo precedente, per l'interpretazione dei dati. Di seguito sono riportati i risultati della prova:

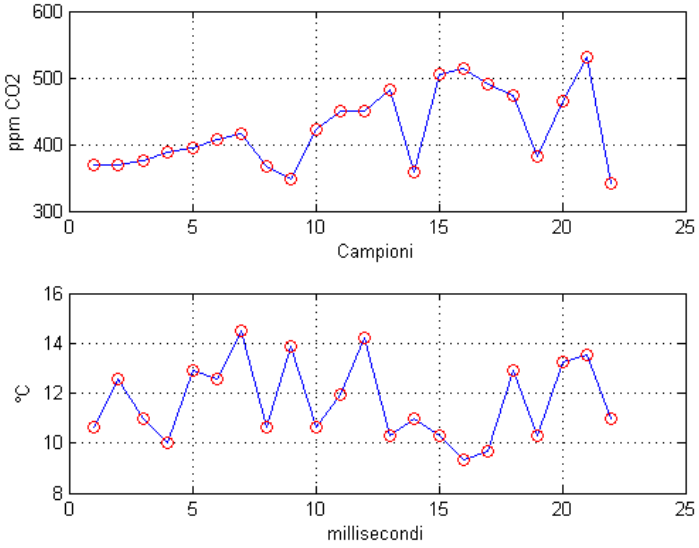


Figura 29: Dati Sensore CO_2 e Temperatura.

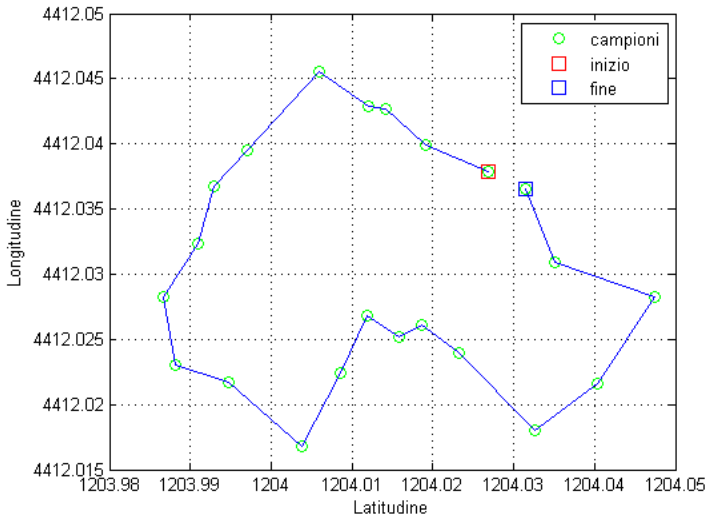


Figura 30: Traccia a terra.

Non vengono riportati i grafici relativi all'andamento dei sensori di CO e di contaminanti dell'aria poichè poco significativi. I dati acquisiti dal sensore di CO_2 rispecchiano i risultati attesi, infatti i livelli di diossido di carbonio si attestano intorno a valori di aria pulita (Figura 29). Lo script del waspmote alterna fasi dedite al riscaldamento di sensori a fasi di acquisizione. Durante il corso della prova, i dati ricevuti dal GPS risultano meno disturbati con il sistema immobile, quindi, in fase di volo, bisogna alternare fasi di avanzamento e fasi di hovering in modo da far avanzare il drone durante il riscaldamento dei sensori e fare acquisizione dati durante la fase di hovering: questa precauzione permette di riceve dati GPS poco rumorosi, causati dalla movimentazione stessa del sistema. In Figura 31 è riportata la traccia a terra di un test effettuato spostando il sistema durante la fase di acquisizione dei dati, seguendo lo stesso percorso in Figura 30. Come si evince è necessario effettuare l'acquisizione dati con il sistema immobile. Ovviamente prima di effettuare ogni test o prova si consiglia di effettuare una calibrazione oculata di tutti i sensori, in quanto i valori di offset sono soggetti a variazioni dovute a cambiamenti di umidità e temperatura. Effettuare questa operazione evita, in fase di analisi dati, di ottenere misure errate nella conversione dei dati acquisiti.

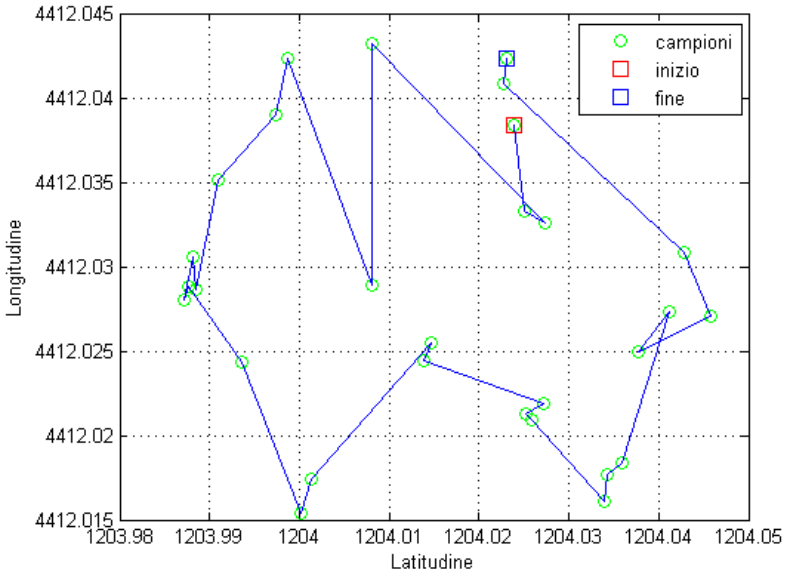


Figura 31: Traccia a terra disturbata.

3.4 Acquisizione dati in volo

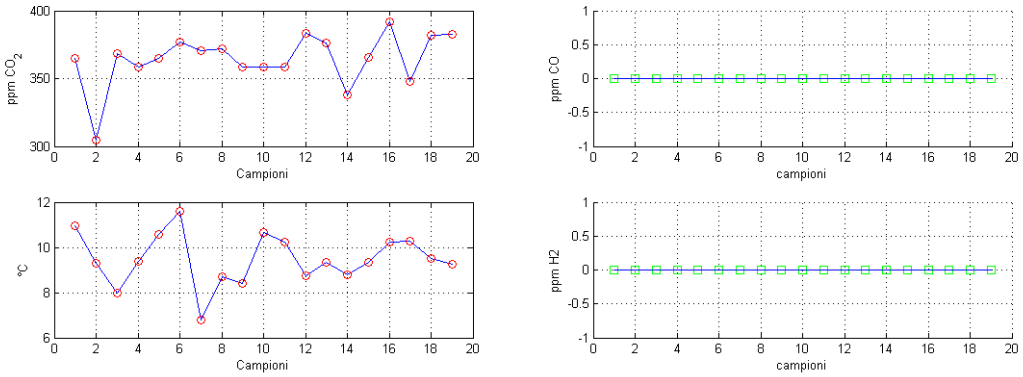
L'obiettivo dell'esperimento prevedeva di mappare il territorio misurando le concentrazioni di gas inquinanti, utilizzando un sistema dinamico, quale è un drone. Al fine di rispettare il regolamento ENAC sull'utilizzo dei SAPR, la prova di volo è stata effettuata nell'aviosuperfice di Villafranca. Il programma utilizzato per l'acquisizione dati (Appendice A) necessita per ogni acquisizione di effettuare un ciclo di riscaldamento dei sensori della durata di 35 secondi. Per evitare di collezionare dati rumorosi, si è deciso di movimentare il drone alternando fasi di avanzamento a fasi di hovering. La manovra di avanzamento viene eseguita durante il riscaldamento dei sensori, mentre il volo a punto fisso durante l'acquisizione dei dati. Il *Waspmote* è stato messo in comunicazione con il PC utilizzando *l'XBee Pro*, in modo tale da avere un controllo diretto sullo svolgimento del codice e dei dati acquisiti. Per effettuare un volo a quota costante e rendere più semplice il controllo del mezzo, l'autopilota è stato utilizzato in modalità *Stabilize* e *Loiter*. La prima permette di mantenere un'assetto stabile in caso di disturbi esterni come raffiche di vento. La seconda, invece, grazie all'accoppiamento con il ricevitore

Figura 32: Drone in volo

GPS permette di mantenere la posizione corrente, la rotta ed una quota costante. Inoltre durante tutto l'arco della prova è stata raccolta la telemetria del mezzo utilizzando il programma *"Mission Planner"*. L'esperimento si è articolato in più fasi:

- accensione del *Waspmote*: stabilizzazione della risposte dei sensori ed acquisizione dei satelliti;
- inizializzazione dei sensori del drone necessarri per il volo;
- decollo;
- acquisizione dati a quota costante, alternando fasi di avanzamento e fasi di hovering;
- atterraggio.

I dati collezionati in volo sono stati analizzati con lo script in ambiente *Matlab*, riportato nell'appendice A. Di seguito sono riportati i grafici più rappresentativi dell'esperimento:



(a) Sensore CO_2 e Temperatura.

(b) Sensore CO

Figura 33: Dati sensori CO_2, temperatura, CO.

Come previsto la concentrazione di CO_2 si attesta intorno a livelli nominali di aria pulita. Sia il sensore di CO che quello dei contaminanti dell'aria non hanno rilevato presenza di gas. E' utile ricordare che i livelli di concentrazione cui sono sensibili questi due dispositivi sono molto elevati, pertanto è molto difficile riuscire a misurare queste concentrazioni nell'ambiente.

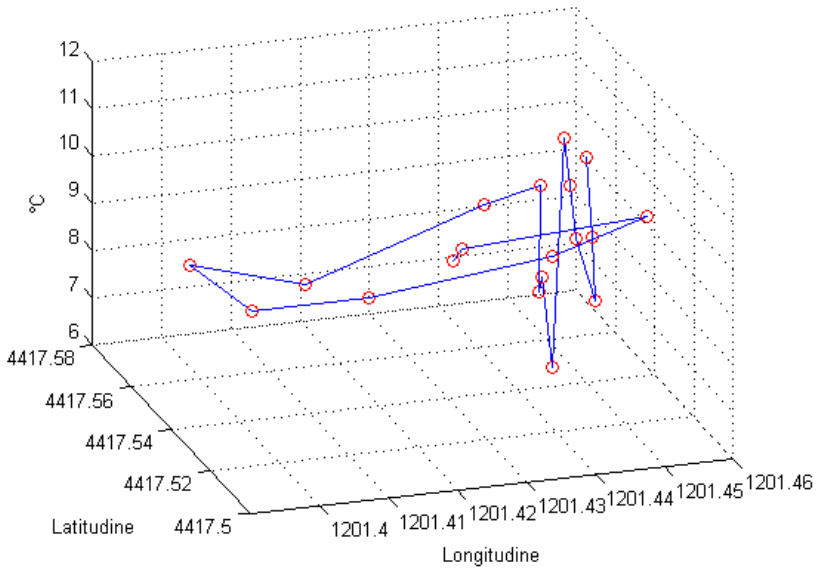
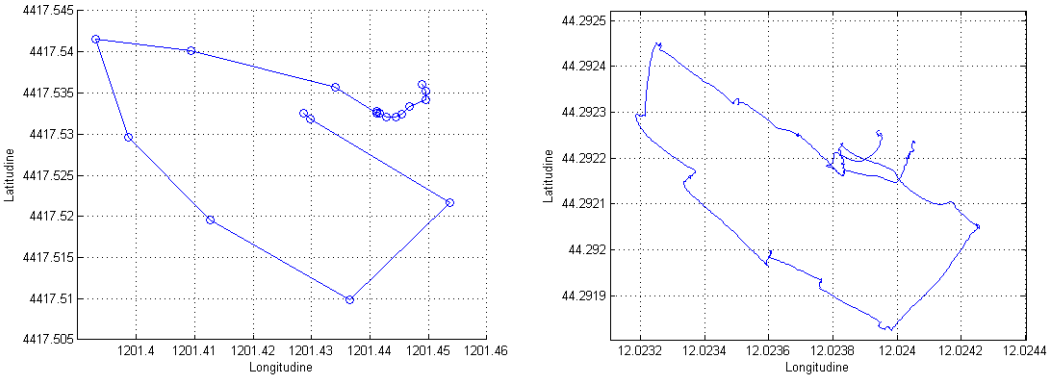


Figura 34: Esempio Mappaura territorio.



(a) GPS Waspmote

(b) GPS Drone

Figura 35: Traccia a terra

Data la poca autonomia oraria di cui dispone il drone utilizzato (10 minuti), è stato possibile acquisire solamente 15 dati utili. Nella Figura 35 (a) e (b) è rappresentato il percorso eseguito utilizzando in (a) i dati GPS del *Waspmote*, e in (b) i dati GPS dell'autopilota. Dato che la board acquisisce un campione ogni 35 secondi, mentre il GPS del drone ne acquisisce 5 al secondo, è normale ottenere una maggior risoluzione come riportato nella Figura 35 (b). I dati acquisiti con il *Waspmote* sono stati sincronizzati tramite le relative coordinate GPS. Un esempio di mappatura del territorio è riportato in Figura 34 dove ad ogni coordinata GPS è stata associata la relativa temperatura misurata. Il grafico in sè non sembra essere così esplicativo, poichè sarebbe opportuno realizzare in ambiente *Google Earth* la mappatura del percorso seguito, associandovi le concentrazioni di gas rilevate. Vale la pena ricordare, inoltre, che il *Waspmote* è stato programmato per il monitraggio della qualità dell'aria, piùttosto che per ottenere precisazioni geografiche. Quindi, i dati che provengono dal GPS della *board*, sepppur non precisi, sono suffiienti per un'analisi preliminare. Per avere una maggiore precisione si consiglia di effettuare un *data fusion* dei dati collezionati con il *Waspmote* e i dati GPS dell'autopilota.

4 Conclusioni

Lo scopo di questo elaborato è quello di realizzare una stazione mobile per il monitoraggio dei gas inquinanti e grazie al prototipo utilizzato è stato possibile conseguire i risultati desiderati. Tuttavia, non va escluso dalle considerazioni il fatto che proprio essendo un prototipo questo tipo di modello può essere continuamente migliorato.
Il set di sensori utilizzato con il *Waspmote* si è rivelato adeguato per una sperimentazione in fase preliminare, durante la quale i sensori sono stati calibrati in maniera approssimativa. Al contrario per una calibrazione più accurata è meglio effettuare il processo in un laboratorio con un ambiente di prova controllato, conoscendo a priori le concentrazioni di gas presenti. In alternativa alla calibrazione si potrebbe utilizzare il set di sensori già calibrati forniti dalla "Libelium".
Altri problemi riscontrati nella fase di sperimentazione riguardano l'assenza di sensori di umidità e pressione. Questi ultimi sono particolarmente importanti perchè permettono di effettuare costanti correzioni sui valori di offset, in maniera tale da evitare una nuova calibrazione ogni qualvolta si utilizzi il sistema a diverse percentuali di umidità e temperatura. Errori sull'impostazione dei valori di offset comportano, infatti, dati sbagliati sulle concentrazioni di gas.
Gli intervalli di rivelabilità dei sensori di CO e contaminanti dell'aria sono molto alti e difficilmente in condizioni ambientali si registrano tali concentrazioni. Dato che i gas a questi livelli sono già nocivi per la salute dell'uomo, i sensori possono essere utilizzati come sensori di allarme, invece di andare a misurare in maniera oculata i vari gas presenti con le relative ppm. A queste considerazioni si può aggiungere che per ottenere una mappattura del territorio più accurata sarebbe opportuno utilizzare un ricevitore GPS più preciso Un'altra alternativa, sarebbe, invece, quella di incrociare i dati acquisiti dal Waspmote con quelli ricevuti dal sistema GPS del drone.
Passando al drone, si denota che quello utilizzato non è propriamente adeguato ad un'acquisizione dati, a causa della bassa autonomia oraria di cui dispone. In una fase sperimentale, come quella presentata in questo elabo-

rato, questo tipo di rilevazione appare più che sufficiente, poichè lo scopo dei test risiede non solo nel monitoraggio della qualità dell'aria, ma anche in quello di osservare il giusto funzionamento del sistema. Con l'intenzione di effettuare lavoro aereo di monitoraggio dell'ambiente, occorrerebbe utilizzare un multirotore con un'autonomia oraria maggiore, in modo tale da avere un raggio di azione più grande. Anche per questo motivo, trova giustificazione l'impiego del drone piuttosto che quello di un monitoraggio da una postazione fissa.

5 Appendice A

5.1 Script *Waspmote* per l'acquisizione dati

```
#include <WaspSensorGas_v20.h>
#include <math.h>
//impostare GAIN e Resistenza di Carico dei Sensori
//CO2
#define GAIN_CO2 7
//CO
#define GAIN_CO 1
#define RESISTOR_CO 100
//AIR_POLL
#define GAIN_AIR_POL 1
#define RESISTOR_AIR_POL 100

//DICHIARAZIONE DELLE VARIABILI
char campioni[100];

//GPS variabili
char fine[5] = ".TXT";
char* filename;//massima lunghezza 9 caratteri
char* DATA_GPS;
char* TIME_GPS;
char* TIME_RTC;
char* LONGITUDE;
char* LATITUDE;
char* SPEED;
char*ALTITUDE;

char coord[100];
char latitude_str[15];
char longitude_str[15];
char speed_str[15];
char altitude_str[15];
char time_gps_str[15];


//Temperatura
float value_T;
char str_T[10];

//CO2
float value_CO2;
char str_CO2[10];
```

```
//CO
float value_CO;
float resistance_CO;
char str_CO[10];
char str_res_CO[10];

//AIR_POL
float value_AIR_POL;
float resistance_AIR_POL;
char str_AIR_POL[10];
char str_res_AIR_POL[10];

//XBee Pro
uint8_t  PANID[2]={0x12,0x34};
char*  KEY="WaspmoteKey";



void setup() {

  //************************** RADIO**********************
  // Inits the XBee 802.15.4 library
  xbee802.init(XBEE_802_15_4,FREQ2_4G,PRO);

  // Powers XBee
  xbee802.ON();

  // Chosing a channel : channel 0x0D
  xbee802.setChannel(0x0D);
  if( !xbee802.error_AT ) XBee.println("Channel set OK");
  else XBee.println("Error while changing channel");

  // Chosing a PANID : PANID=0x1234
  xbee802.setPAN(PANID);
  if( !xbee802.error_AT ) XBee.println("PANID set OK");
  else XBee.println("Error while changing PANID");

  // Enabling security : KEY="WaspmoteKey"
  xbee802.encryptionMode(1);
  if( !xbee802.error_AT ) XBee.println("Security enabled");
  else XBee.println("Error while enabling security");
```

```
xbee802.setLinkKey(KEY);
if( !xbee802.error_AT ) XBee.println("Key set OK");
else XBee.println("Error while setting Key");

// Keep values
xbee802.writeValues();
if( !xbee802.error_AT ) XBee.println("Changes stored OK");
else XBee.println("Error while storing values");

//-----------------------------------------------------


USB.begin();
XBee.println("start");

XBee.print("Battery Level: ");
XBee.print(PWR.getBatteryVolts());
XBee.println(" Volts");


SD.ON();
XBee.println("accensione SD");

XBee.println("Setting Up GPS...");
GPS.init(); //inizializzazione Gps
GPS.ON();

 //agganciare satelliti
do {
  GPS.check();
  XBee.println("ricerca satelliti in corso...");
  delay(5000);
}
while(!GPS.check());

XBee.println("satelliti agganciati");
GPS.getTime();
GPS.getDate();

//impostarre il nome del file con ora e data GPS
DATA_GPS = GPS.getDate(); //"ddmmyy"
TIME_GPS = GPS.getTime(); //"hhmmss.mmmmm"
```

```
  sprintf(filename,"%c%c%c%c%c%c%c%c", DATA_GPS[0],
                                       DATA_GPS[1],
                                       DATA_GPS[2],
                                       DATA_GPS[3],
                                       TIME_GPS[0],
                                       TIME_GPS[1],
                                       TIME_GPS[2],
                                       TIME_GPS[3]
                                       );

  strcat(filename,fine);

  XBee.print("Nome del FILE: ");
  XBee.println(filename);



  if(SD.create(filename))
  {
    XBee.print("FILE CREATO: ");
    XBee.println(filename);
  }
  else XBee.println("File NON Creato");
  SD.append(filename,"Ore,Minuti,Secondi,Latitudine,Longitudine,Speed,Altitudine");
  SD.appendln(filename,"TEMPERATURA [Celsius],CO2[VOLT],CO[VOLT],");
  SD.appendln(filename,"Res_CO[KOhm],AIR_POL[Volt],AIR_POL[KOhm]");
}

  void loop(){
 XBee.println("attendi 30 secondi per heating Gases Board");
    ////////////////////////////
    //ACCENSIONE GASES BOARD
    ////////////////////////////

  //prova 1.0
    //SensorGasv20.setBoardMode(SENS_ON); //accendo la Gases Board
  //prova 1.1
    SensorGasv20.ON();

    //NB prima di accendere i sensori devi configurarli
    SensorGasv20.configureSensor(SENS_CO2,GAIN_CO2);//CO2
    SensorGasv20.configureSensor(SENS_SOCKET3CO,GAIN_CO,RESISTOR_CO);//CO
```

```
  SensorGasv20.configureSensor(SENS_SOCKET4A,GAIN_AIR_POL,RESISTOR_AIR_POL);//AIR_POL

  SensorGasv20.setSensorMode(SENS_ON,SENS_CO2);//CO2
  SensorGasv20.setSensorMode(SENS_ON,SENS_SOCKET3CO);//CO
  SensorGasv20.setSensorMode(SENS_ON,SENS_SOCKET4A);//AIR_POL
  delay(30000);

 //acquisizione dati dai sensori
 value_T = SensorGasv20.readValue(SENS_TEMPERATURE);
 value_CO2 = SensorGasv20.readValue(SENS_CO2);
 value_CO = SensorGasv20.readValue(SENS_SOCKET3CO);
 value_AIR_POL= SensorGasv20.readValue(SENS_SOCKET4A);

  //invio dati alla Ground Station
  XBee.println("-------------------");
  XBee.print("temperatura: ");
  XBee.print(value_T);
  XBee.println(" celsius");

  XBee.print("Vout CO2: ");
  XBee.print(value_CO2);
  XBee.println(" Volt");

  XBee.print("volt CO: ");
  XBee.print(value_CO);
  XBee.println(" Volts");

  XBee.print("volt AIR POL: ");
  XBee.print(value_AIR_POL);
  XBee.println(" Volts");
  XBee.println("-------------------");

 ////////////////////////////////////////////////////
 //SPEGNERE GASES BOARD per raffreddamento sensori
 ////////////////////////////////////////////////////
  SensorGasv20.OFF();
'////////////////////////////////////////////////////
'////////////////////////////////////////////////////

 //Calcolo della resistenza equivalente per sensore CO e contaminanti dell'aria
 resistance_CO = calcola_res(RESISTOR_CO,value_CO);
 resistance_AIR_POL=calcola_res(RESISTOR_AIR_POL,value_AIR_POL);
```

```
  //temperatura
  Utils.float2String(value_T, str_T,5);
  //CO2
  Utils.float2String(value_CO2, str_CO2,5);
  //CO
  Utils.float2String(value_CO, str_CO,5);
  Utils.float2String(resistance_CO, str_res_CO,5);
  //AIR_POL
  Utils.float2String(value_AIR_POL, str_AIR_POL,5);
  Utils.float2String(resistance_AIR_POL, str_res_AIR_POL,5);


  sprintf(campioni,"%s,%s,%s,%s,%s,%s",str_T,str_CO2,str_CO,
                                       str_res_CO,str_AIR_POL,
                                       str_res_AIR_POL);
 ////////////////////////////////////////////////////////////////
 //////GPS
 ////////////////////////////////////////////////////////////////
//acquisizione dati dal GPS
     if (GPS.getPosition())
  {
    TIME_GPS = GPS.getTime(); // "hhmmss"
    LATITUDE = GPS.getLatitude();
    LONGITUDE = GPS.getLongitude();
    SPEED = GPS.getSpeed();
    ALTITUDE= GPS.getAltitude();


    //strcpy(tempo_str,TIME_RTC);
    sprintf(time_gps_str,"%c%c,%c%c,%c%c", TIME_GPS[0],
                                           TIME_GPS[1],
                                           TIME_GPS[2],
                                           TIME_GPS[3],
       |                                   TIME_GPS[4],
                                           TIME_GPS[5]);

    strcpy(latitude_str, LATITUDE);
    strcpy(longitude_str,LONGITUDE);
    strcpy(speed_str,SPEED);
    strcpy(altitude_str,ALTITUDE);
```

```
    XBee.println("---------------");
    XBee.println(time_gps_str);
    XBee.println(latitude_str);
    XBee.println(longitude_str);
    XBee.println(speed_str);
    XBee.println(altitude_str);
    XBee.println("---------------");

sprintf(coord,"%s,%s,%s,%s,%s,",time_gps_str,
                                        latitude_str,
                                        longitude_str,
                                        speed_str,
                                        altitude_str);

//scrittura file coordinate
        if(SD.append(filename,coord))
     {
       XBee.println("vettore scritto nel file:");
       XBee.println(coord);
     }
       else XBee.println("errore nella scrittura su SD");
     }
      else XBee.println("errore nella compilazione del file");
//scrittura file sensori
   if(SD.appendln(filename,campioni)){
    XBee.print("dato scritto:");
    XBee.println(campioni);
  }
  else XBee.println("errore scrittura nel file ");
  delay(3000);
}
//funzione che calola la resistenza dei sensori
float calcola_res (float load_sens, float tensione)
{
  float resistenza;

  resistenza = (5*load_sens)/tensione - load_sens;
  return (resistenza);
}
```

5.2 Script Analisi Dati MATLAB

```
clc
close all
load 'Prova_finale_tesi.mat'
millis=linspace(1,length(Res_CO),length(Res_CO))';
%%variabili CO2
GAIN_CO2 = 7;
CO2_offset=300.2151;
new_CO2=(CO2_value./GAIN_CO2).*1000;
y=64.305*log10(x)-163.596;
%coversione CO2 da volt a ppm
delta_EMF=CO2_offset-new_CO2;
ppm_CO2=10.^((delta_EMF+163.596)./64.305);
figure(1)
title('TGS 4161 CO2, datasheet')
hold on
xlabel('ppm_CO2');
ylabel('deltaEMF');
plot(x,y);
set(gca ,'XScale','log')
axis([10,10000, 0,100]);
grid on
plot(3000,60,'-sr');
plot(350,0,'-sr');
plot(1000,29.319,'-sr');
plot(10000,93.624,'-sr');
legend('andamento sensore','punti noti datasheet');
%Analisi dati CO2 TGS 2141
figure(2)
title('Dati_statici')
hold on
plot(x,y);
plot(ppm_CO2,delta_EMF,'or');
set(gca ,'XScale','log');
axis([10,10000, 0,100]);
xlabel('ppm_CO2');
ylabel('delta EMF');
legend('datasheet','campioni acquisiti');
grid on
```

```
figure(3)
subplot(211)
plot(millis,ppm_CO2);
hold on
plot(millis,ppm_CO2,'or');
ylabel('ppm CO_2');
xlabel('Campioni');
grid on
subplot(212)
plot(millis,TEMPERATURA);
hold on
plot(millis,TEMPERATURA,'or');
xlabel('Campioni');
ylabel('°C');
grid on
%%sensore CO
R0_CO = 13;%offset
Rs_R0_CO= Res_CO./R0_CO;
k=1;
j=1;
m=1;
H2_ohm=zeros(length(Rs_R0_CO),2);
H2_ohm(:,2)=millis(:,1)
CO_new=zeros(length(Rs_R0_CO),2);
length(Rs_R0_CO);
for i=1:length(Rs_R0_CO)
                    if  (Rs_R0_CO(i,1) > 5) && (Rs_R0_CO(i,1) < 10)
                    H2_ohm(k,1)= Rs_R0_CO(i,1);
                    H2_ohm(k,2)= millis(i,1);
                    k=k+1;

                    elseif  ( Rs_R0_CO(i,1)>0.08) && (Rs_R0_CO(i,1)<4)
                        CO_new(j,1) = Rs_R0_CO(i,1);
                        CO_new(j,2) = millis(i,1);
                        j=j+1;
                    else
                        CO_pulita(m,1)= Rs_R0_CO(i,1);
                        CO_pulita(m,2)= millis(i,1);
                        m=m+1;
                    end
end
```

```
ppm_CO = zeros(length(Rs_R0_CO),2)
ppm_H2 = zeros(length(Rs_R0_CO),2)
 %ppm_H2
 if (H2_ohm(:,1)~= 0)
 ppm_H2=10.^((-log10(H2_ohm(:,1))+2.475)./0.592);
 end
 if (CO_new(:,1)~= 0)
%%ppm_CO
for i=1:length(CO_new)

          if (CO_new(i,1)>1)&&(CO_new(i,1)<4)
           ppm_CO(i,1)= 10.^((-log10(CO_new(i,1))+2.302)./1.151);
           ppm_CO(i,2)= CO_new(i,2);

          elseif (CO_new(i,1)>0.25)&&(CO_new(i,1)<1)
           ppm_CO(i,1)= 10.^((-log10(CO_new(i,1))+2.520)./1.262);
           ppm_CO(i,2)= CO_new(i,2);

          else (CO_new(i,1)>0.08)&&(CO_new(i,1)<0.25)
           ppm_CO(i,1)=10.^((-log10(CO_new(i,1))+1.741)./0.946);
           ppm_CO(i,2)= CO_new(i,2);

          end

end
 end
 CO_P=[30 4; 100 1; 300 0.25; 1000 0.08];
 H2_P=[300 10.2 ; 1000 5];
%%datasheet CO
 figure(4)
 title('TGS 2442 - CO')
 hold on
 plot(CO_P(:,1),CO_P(:,2),'b');
 plot(CO_P(:,1),CO_P(:,2),'or');
 plot(H2_P(:,1),H2_P(:,2),'k');
 plot(H2_P(:,1),H2_P(:,2),'sk');
 set(gca,'XScale','log','YScale','log');
 axis([1,10000,0.01,100]);
 xlabel('Concentrazione Gas (ppm)');
 ylabel('Rs/R0 [KOhm]');
 grid
```

```
%caso in cui o i ppm_CO o i ppm_H2 siano diversi da zero
if((CO_new(1,1)~= 0)||(CO_new(1,1)~= 0))
figure(5)
subplot(311)
hold on
plot(CO_P(:,1),CO_P(:,2),'b');
plot(H2_P(:,1),H2_P(:,2),'k');
plot(ppm_CO(:,1),CO_new(:,1),'sg');
set(gca,'XScale','log','YScale','log');
xlabel('Concentrazione Gas (ppm)');
ylabel('Rs/R0 [KOhm]');
%axis([1,1000,0.07,20]);
grid
subplot(211)
hold on
plot(ppm_CO(:,2),ppm_CO(:,1),'sg');
plot(ppm_CO(:,2),ppm_CO(:,1),'b');
xlabel('tempo (millisecondi)');
ylabel('ppm CO');
grid
subplot(212)
hold on
plot(ppm_H2(:,2),ppm_H2(:,1));
xlabel('tempo (millisecondi)');
ylabel('ppm H2');
grid
end
if (CO_pulita(1,1)~=0)
figure(5)
subplot(211)
hold on
plot(millis,0,'sg');
plot(millis,zeros(length(millis),1),'b');
xlabel('campioni');
ylabel('ppm CO');
grid
subplot(212)
plot(millis,0,'sg');
plot(millis,zeros(length(millis),1),'b');
xlabel('campioni');
ylabel('ppm H2');
end
```

```
%%dividere campioni per quota
for i=1:length(Altitudine)
    if (Altitudine(i)>fix(Altitudine(i))-0.25)&&
        (Altitudine(i)<fix(Altitudine(i))+0.25)
        Altitudine(i)=fix(Altitudine(i));

    elseif (Altitudine(i)>fix(Altitudine(i))+0.25)&&
           (Altitudine(i)<fix(Altitudine(i))+0.75)
            Altitudine(i)=fix(Altitudine(i))+0.5;

    else (Altitudine(i)>fix(Altitudine(i))+0.75)&&
        (Altitudine(i)<fix(Altitudine(i))+1.25)
          Altitudine(i)=fix(Altitudine(i))+1;
    end
end
%%quota1=quota'

%%ripulire la matrice con valori medi dei sensori
for i=1:length(Altitudine(:,1))
    k=0;
    mid_temp=0;
    mid_ppm_CO2=0;
    mid_ppm_H2=0;
    mid_ppm_CO=0;

    for j=1:length(Altitudine(:,1))

        if(isequal(Altitudine(i),Altitudine(j)))
           k=k+1;
           mid_temp = mid_temp + TEMPERATURA(j);
           mid_ppm_CO2 = mid_ppm_CO2 + ppm_CO2(j,1);
           mid_ppm_H2 = mid_ppm_H2 + ppm_H2(j,1);
           mid_ppm_CO = mid_ppm_CO + ppm_CO(j,1);
        end
    end
    new_campioni(i,1)=Altitudine(i); %quota
    new_campioni(i,2)=mid_temp./k; %temperatura
    new_campioni(i,3)=mid_ppm_CO2./k; %ppm_CO2
    new_campioni(i,4)=mid_ppm_H2./k;%ppm_H2 sensore CO
    new_campioni(i,5)=mid_ppm_CO./k;%ppm CO sensore CO
end
```

```
%%elimino le celle
campioni_compatti=unique(new_campioni,'rows');
figure
subplot(141)
barh(campioni_compatti(:,1),campioni_compatti(:,2));
xlabel('°C')
ylabel('Altitudine (m)')
grid
subplot(142)
barh(campioni_compatti(:,1),campioni_compatti(:,3));
grid
ylabel('ppm CO2')
subplot(143)
barh(campioni_compatti(:,1),campioni_compatti(:,4));
grid
ylabel('ppm H2')
subplot(144)
barh(campioni_compatti(:,1),campioni_compatti(:,5));
ylabel('ppm CO')
xlabel('altitudine m');
grid
figure
hold on
plot(Longitudine,Latitudine);
hold on
plot(Longitudine,Latitudine,'o');
xlabel('Longitudine')
ylabel('Latitudine')
grid
figure
plot3(Longitudine,Latitudine,TEMPERATURA,'or')
hold on
plot3(Longitudine,Latitudine,TEMPERATURA)
hold on
xlabel('Longitudine')
ylabel('Latitudine')
zlabel('°C')
view(-30,10)
grid
```

Riferimenti bibliografici

[1] Libelium Comunicaciones Distribuidas S.L. (2012), *Waspmote Technical Guide.*

[2] Libelium Comunicaciones Distribuidas S.L. (2012), *Gases Sensor Board 2.0 Technical Guide.*

[3] FIGARO (2012), *TGS 4161 Product information.*

[4] FIGARO (2007), *TGS 2442 Product Information.*

[5] FIGARO (2005), *TGS 2602 Product Information.*

[6] Microchip Technology Inc. (2014), *MCP9700A Product Information.*

[7] Alice Zaccone (2012), *Studio e sperimentazione di una scheda per l'acquisizione di concentrazioni di sostanze presenti in atmosfera*, Elaborato finale di Laurea, Università di Bologna (Forlì).

[8] Massimiliano Menghini (2015), *Studio del sistema embedded "Waspmote – Gases Board" per il monitoraggio della qualità dell'aria*, Elaborato finale di tirocinio curriculare, Università di Bologna (Forlì).