

POLITECNICO DI MILANO  
V Facoltà di Ingegneria  
Corso di Laurea Specialistica in Ingegneria delle Telecomunicazioni  
Dipartimento di Elettronica e dell'Informazione



# Sviluppo di un sistema per il monitoraggio ambientale basato su reti di sensori

Relatore: prof. Antonio CAPONE

Tesi di Laurea di:  
Domenico SCHILLACI Matr. 719689

Anno Accademico 2009 / 2010



Dedicato a mio zio Andrea

# Indice

<b>Introduzione</b>	<b>1</b>
<b>1 Un dispositivo per misurazioni distribuite a metà tra una rete di sensori e un sistema embedded</b>	<b>5</b>
1.1 Wireless Sensor Networks . . . . .	5
1.1.1 Definizione . . . . .	5
1.1.2 Caratteristiche . . . . .	6
1.1.3 Applicazioni . . . . .	9
1.1.3.1 Applicazioni militari . . . . .	9
1.1.3.2 Applicazioni ambientali . . . . .	9
1.1.3.3 Applicazioni medico-sanitarie . . . . .	10
1.1.3.4 Applicazioni domestiche . . . . .	10
1.1.3.5 Applicazioni commerciali . . . . .	10
1.1.4 Architettura di comunicazione . . . . .	11
1.1.4.1 Livello di applicazione . . . . .	12
1.1.4.2 Livello di rete . . . . .	12
1.1.4.3 Livello MAC . . . . .	13
1.1.5 Progetto di rete . . . . .	14
1.1.5.1 Tolleranza ai guasti . . . . .	15
1.1.5.2 Scalabilità . . . . .	15
1.1.5.3 Costi . . . . .	15
1.1.5.4 Ambiente operativo . . . . .	16
1.1.5.5 Topologia di rete . . . . .	16
1.1.5.6 Vincoli hardware . . . . .	17
1.1.5.7 Mezzo trasmissivo . . . . .	17

1.2	Sistemi embedded . . . . .	18
1.2.1	Definizione . . . . .	18
1.2.2	Caratteristiche . . . . .	19
1.2.2.1	Specializzazione . . . . .	19
1.2.2.2	Hardware dedicato . . . . .	19
1.2.2.3	Applicazioni real-time . . . . .	20
1.2.2.4	Sistemi distribuiti . . . . .	20
1.2.3	Applicazioni . . . . .	21
1.2.4	Requisiti di progetto . . . . .	22
1.2.4.1	Requisiti funzionali . . . . .	22
1.2.4.2	Requisiti temporali . . . . .	23
1.2.4.3	Requisiti di affidabilità . . . . .	23
1.2.4.4	Consumi . . . . .	23
1.2.4.5	Prestazioni . . . . .	23
1.2.4.6	Costi . . . . .	24
1.3	Sistemi di misura distribuiti . . . . .	25
1.3.1	Il concetto di misurazione distribuita . . . . .	25
1.3.2	Architettura di comunicazione . . . . .	25
1.3.3	Campi di applicazione e sviluppo . . . . .	26
<b>2</b>	<b>Idee, motivazioni e prospettive di sviluppo alla base del progetto FLOWEATHER</b>	<b>27</b>
2.1	Idee . . . . .	27
2.1.1	Analisi dei dati meteorologici e relativi all'inquinamento . . . . .	27
2.1.2	Integrazione in rete di diversi dispositivi e mappa climatica . . . . .	27
2.1.3	Presentazione tramite web e personalizzazione del servizio . . . . .	28
2.1.4	Flessibilità da due punti di vista: modularità e mobilità . . . . .	29
2.1.5	Consumi energetici ridotti . . . . .	29
2.1.6	Design industriale: la scelta del vaso da fiori . . . . .	29
2.2	Motivazioni . . . . .	30
2.2.1	Un metodo innovativo per misurazioni vicine all'utente . . . . .	30
2.2.2	Dati meteorologici orari e mappa climatica dei centri urbani . . . . .	30
2.2.3	Uno strumento per la gestione energetica degli edifici . . . . .	32

2.2.4	Il monitoraggio ambientale dal punto di vista dell'inquinamento . . . . .	33
2.2.5	Originalità e innovatività . . . . .	33
2.3	Prospettive di sviluppo . . . . .	34
2.3.1	Commercializzazione del prodotto: l'ambito consumer . . . . .	34
2.3.2	Un secondo possibile utilizzo: l'analisi scientifica . . . . .	34
<b>3</b>	<b>Progettazione e realizzazione del prototipo</b>	<b>36</b>
3.1	Definizione dei driver di progetto . . . . .	36
3.1.1	La prospettiva <i>Green</i> . . . . .	36
3.1.2	La prospettiva <i>Low cost</i> . . . . .	37
3.1.3	La prospettiva <i>Open Source</i> . . . . .	37
3.1.4	La prospettiva <i>Sustainable</i> . . . . .	38
3.1.5	Una doppia direttrice di sviluppo: <i>floweatherpot</i> e <i>floweather.com</i> . . . . .	38
3.2	Il dispositivo: <i>floweatherpot</i> . . . . .	39
3.2.1	Caratteristiche . . . . .	39
3.2.1.1	Mobilità . . . . .	39
3.2.1.2	Modularità . . . . .	39
3.2.1.3	Hardware libero . . . . .	40
3.2.1.4	Flessibilità . . . . .	40
3.2.2	Componenti . . . . .	41
3.2.2.1	Scheda Arduino Duemilanove . . . . .	41
3.2.2.2	Sensori . . . . .	42
3.2.2.3	Modulo GSM/GPRS . . . . .	44
3.2.2.4	Real time clock . . . . .	45
3.2.2.5	Display LCD . . . . .	45
3.2.2.6	Batterie e circuito di ricarica . . . . .	46
3.2.3	Funzionalità . . . . .	48
3.2.3.1	Sensing . . . . .	48
3.2.3.2	Comunicazione . . . . .	49
3.2.3.3	Controllo dei consumi . . . . .	50
3.2.4	Progetto . . . . .	51
3.2.4.1	Schema elettrico . . . . .	51

3.2.4.2	Codice sorgente . . . . .	52
3.2.4.3	Tabella dei consumi . . . . .	60
3.2.4.4	Tabella dei costi . . . . .	61
3.2.5	Componenti aggiuntivi . . . . .	62
3.2.5.1	Pannello solare . . . . .	62
3.2.5.2	Anemometro . . . . .	62
3.2.5.3	Board di sensori per la qualità dell'aria . . . . .	62
3.2.5.4	Board di sensori per l'inquinamento acustico . . . . .	63
3.2.5.5	Modulo di comunicazione basato sul protocollo ZigBee . . . . .	63
3.2.6	Design . . . . .	64
3.2.6.1	Materiale . . . . .	65
3.2.6.2	Progetto . . . . .	65
3.3	La pagina web: <i>floweather.com</i> . . . . .	67
3.3.1	Caratteristiche . . . . .	67
3.3.1.1	Semplicità . . . . .	67
3.3.1.2	Personalizzazione . . . . .	68
3.3.1.3	Aggiornamento in tempo reale . . . . .	68
3.3.1.4	Automatizzazione . . . . .	68
3.3.2	Funzionalità . . . . .	68
3.3.2.1	Memorizzazione . . . . .	69
3.3.2.2	Presentazione . . . . .	72
3.3.2.3	Utilizzo all'interno del social network Twitter . . . . .	76
3.3.2.4	Utilizzo all'interno di software di simulazione energetica . . . . .	76
	<b>Conclusioni</b>	<b>77</b>
	<b>A Appendice</b>	<b>80</b>
	<b>Ringraziamenti</b>	<b>88</b>

# Elenco delle figure

1	Render del progetto . . . . .	viii
2	Logo del progetto . . . . .	4
1.1	Schema di un generico nodo sensore . . . . .	5
1.2	Schema di una generica rete di sensori . . . . .	6
1.3	Trasmissione dell'informazione dal nodo attraverso il sink fino all'utente finale . . . . .	8
1.4	Modalità di monitoraggio di un'area attraverso l'utilizzo di sensori . . . . .	8
1.5	Esempio di trasmissione di informazioni dal paziente al medico per mezzo di WSN . . . . .	10
1.6	Stack protocollare di una rete di sensori . . . . .	11
1.7	Schema di un generico sistema embedded . . . . .	20
1.8	Esempio di applicazione embedded . . . . .	21
1.9	Esempio di sistema di misura distribuito per la prevenzione degli incendi . . . . .	24
2.1	Topologia a stella della rete di sensori . . . . .	28
2.2	Schermata di Climate Consultant 4: grafico di ombreggiatura solare ricavato a partire ai dati climatici orari raccolti nella zona di Milano Linate nel 2009 . . . . .	31
2.3	Schermata di Climate Consultant 4: Consigli sulle strategie di progetto ricavati in base ai dati climatici orari misurati nella zona di Milano Linate nel 2009 . . . . .	32
3.1	Scheda Arduino con evidenziati i diversi componenti . . . . .	42
3.2	Sensore SHT15 . . . . .	43
3.3	Sensore SHT15 . . . . .	43
3.4	Sensore TGS2442 . . . . .	44
3.5	Modulo GM862 . . . . .	44
3.6	RTC BV4236 . . . . .	45
3.7	Display LCD . . . . .	46
3.8	Pack di batterie agli ioni di litio . . . . .	46

3.9	Schema di progetto . . . . .	64
3.10	Prospettiva frontale . . . . .	66
3.11	Prospettiva laterale destra . . . . .	66
3.12	Sezione laterale . . . . .	66
3.13	Schermata web: <a href="http://floweather.com/myfloweather/login">floweather.com/myfloweather/login</a> . . . . .	73
3.14	Schermata web: <a href="http://floweather.com/myfloweather/station">floweather.com/myfloweather/station</a> . . . . .	73
3.15	Schermata web: <a href="http://floweather.com/myfloweather/location">floweather.com/myfloweather/location</a> . . . . .	74
3.16	Schermata web: <a href="http://floweather.com/myfloweather/hTem">floweather.com/myfloweather/hTem</a> . . . . .	74
3.17	Schermata web: <a href="http://floweather.com/myfloweather/dmTem">floweather.com/myfloweather/dmTem</a> . . . . .	75
3.18	Schermata web: <a href="https://twitter.com/floweather">twitter.com/floweather</a> . . . . .	75

# Elenco delle tabelle

3.1	Panoramica dei componenti impiegati . . . . .	47
3.2	Componenti e relativi consumi a seconda dei modi di funzionamento . . . . .	60
3.3	Componenti e relativi costi . . . . .	61



Figura 1: Render del progetto

# Introduzione

La necessità di una conoscenza approfondita dell'ambiente all'interno del quale siamo immersi quotidianamente, unita alla capacità di esercitare un controllo costante su quelle variabili che possono apparire critiche per la salute ed il benessere dell'uomo, rappresentano oggi temi di grande attualità.

Questa crescente esigenza di relazionarsi con l'ambiente circostante ha portato, negli ultimi anni, alla creazione di particolari sistemi studiati appositamente per il monitoraggio territoriale, che prendono il nome di sistemi di misura distribuiti. Costituiti da un insieme di dispositivi in grado di effettuare complesse operazioni di misurazione, vengono posizionati in punti differenti dell'area da analizzare e possono comunicare tra loro per mezzo di specifici standard per la trasmissione delle informazioni. Il vantaggio fondamentale portato dall'adozione di questo tipo di sistemi consiste nella possibilità di mantenere sotto controllo l'intero ambiente da diversi punti di vista, pur trovandosi a notevole distanza da esso, e di aggregare i dati raccolti dalle diverse postazioni in modo da ottenere una visione più completa e sistematica della realtà presa in esame.

Per realizzare applicazioni di questo genere si è soliti orientarsi sulla tecnologia delle reti di sensori wireless, che, essendo basata su un'architettura distribuita, permette agli elementi che la compongono di operare un rilevamento rispetto a specifiche grandezze fisiche e di scambiarsi i risultati di tale processo. L'elemento principale di questa tipologia di rete è costituito dal nodo, altresì chiamato nodo sensore o semplicemente sensore, un dispositivo di piccole dimensioni e dai costi ridotti capace di lavorare a basse potenze e dialogare con i suoi simili tramite un protocollo di comunicazione a corto raggio.

L'innovazione tecnologica che ha investito negli ultimi anni il campo delle telecomunicazioni e più in generale dell'elettronica ha portato ad un graduale aumento della complessità dei sensori oltre che ad un incremento delle funzionalità supportate, che ha prodotto a sua volta una migrazione verso la creazione di dispositivi autosufficienti, estremamente funzionali e progettati per scopi specifici. Questa trasformazione ha contribuito fortemente ad avvicinare il concetto di nodo di rete a quello di sistema embedded, che identifica quei dispositivi elettronici programmati per una particolare applicazione e non riconfigurabili in seguito.

Una buona strategia di sviluppo per un dispositivo studiato con l'intento di compiere misure distribuite dovrebbe quindi porsi a metà strada tra le due suddette tecnologie, cercando di cogliere gli aspetti chiave, ora delle reti di sensori ora dei sistemi embedded.

Il lavoro proposto in questo elaborato mira, attraverso l'implementazione di un dispositivo, a fornire un approccio innovativo in merito al tema delle misurazioni distribuite, evidenziando al contempo le soluzioni più adatte per realizzare un servizio di monitoraggio dell'ambiente che sia vicino all'utente.

A partire da questi presupposti nasce il progetto FLOWEATHER, che si concentra sull'analisi di due particolari tipologie di misura: da un lato i dati meteorologici (o climatici), dall'altro quelli relativi all'inquinamento ambientale (atmosferaico, acustico, elettromagnetico). Esso prevede la fabbricazione, sotto forma di prototipo, di una stazione mobile in grado di studiare il comportamento dello spazio che la circonda, tenendolo costantemente sotto osservazione. I dati raccolti, una volta immagazzinati, vengono trasmessi, tramite una connessione internet, ad un database che li memorizza in maniera strutturata e provvede a presentarli agli utenti. La visualizzazione delle misure collezionate nel tempo avviene grazie ad una pagina web che si occupa di erogare un servizio personalizzato basato sull'impiego di strutture grafiche come diagrammi e tabelle riassuntive.

La possibilità di integrare un certo numero di dispositivi all'interno di una precisa area di interesse, come ad esempio un centro urbano, rappresenta una delle maggiori potenzialità del progetto, in quanto consentirebbe di stilare, a partire dai dati accumulati dalle diverse stazioni poste sul territorio, una mappa globale del clima o della qualità dell'aria, operando di conseguenza delle valutazioni che altrimenti non si sarebbe in alcun modo in grado di fare.

I due tratti maggiormente distintivi del progetto sono rappresentati dalla flessibilità e dai consumi energetici ridotti. Il primo aspetto è ottenuto a partire dalle caratteristiche di mobilità e modularità che conferiscono una libertà assoluta sia in termini di posizionamento della stazione sia di possibili espansioni future legate all'implementazione di moduli aggiuntivi.

La sostenibilità energetica appare dal canto suo di fondamentale importanza poiché permette di allungare il periodo di vita del dispositivo e, di conseguenza, di migliorarne l'autonomia.

Uno degli obiettivi primari che ha guidato la fase di progettazione del prototipo è stato volto a porre le basi per trasformare il modello sperimentale in un prodotto per il mercato, nell'ottica di una eventuale commercializzazione. Per far ciò è stato necessario curare quei risvolti tipici della produzione industriale quali il costo finale e l'aspetto estetico dell'oggetto da vendere. A tale scopo, trattandosi di una stazione per uso essenzialmente privato, studiata per essere collocata in esterni, si è deciso di conferirle le sembianze di un piccolo vaso da fiori, così da aggiungere alla prospettiva scientifica anche quella decorativa.

FLOWEATHER si presenta perciò come una piattaforma per misurazioni distribuite racchiusa all'interno di un design essenziale, pratico e curato nelle linee.

I motivi che hanno portato alla scelta di progetto di questo tipo sono individuabili nell'originalità ed innovatività dell'approccio adottato, unita alla vasta applicabilità mostrata.

L'innovatività proposta si fonda sull'idea di riuscire ad attribuire ai dati immagazzinati una prerogativa locale, che contribuisca a renderli strettamente legati al territorio in cui essi sono

rilevati. Essa è inoltre rafforzata dal metodo impiegato per la presentazione dei dati, ideato per risultare chiaro e immediatamente comprensibile.

I possibili campi di applicazione spaziano dal monitoraggio del grado di inquinamento atmosferico o acustico nei centri cittadini all'utilizzo dei dati climatici orari raccolti in ambito urbanistico, al fine di migliorare i processi di progettazione e successiva gestione degli edifici dal punto di vista energetico.

La capacità di sfruttare le misurazioni a disposizione all'interno di particolari software per simulazioni energetiche contribuisce inoltre a conferire un valore aggiunto al progetto.

Per quanto concerne la realizzazione pratica del prototipo, è stato inizialmente necessario individuare quei particolari driver a cui ispirarsi durante il lavoro progettuale; essi possono essere ben riassunti dall'acronimo GLOSS (Green, Low cost, Open Source, Sustainable).

Accanto alle prospettive di sostenibilità e contenimento dei costi, di cui si è già accennato, trovano collocazione l'impegno ambientale, testimoniato anche dalla scelta dei componenti, e l'adozione della filosofia open source, che mira alla condivisione per far progredire il progetto.

Il dispositivo è stato quindi costruito assemblando insieme i componenti ritenuti più adatti a garantirne la massima flessibilità. Come microcontrollore è stata impiegata una scheda Arduino Duemilanove, uno dei primi esempi di hardware libero e di produzione italiana. Il rilevamento è stato compiuto prevedendo l'inserimento di tre sensori con lo scopo di misurare quattro grandezze: temperatura, umidità, pressione atmosferica e concentrazione di monossido di carbonio nell'aria. Per la comunicazione è stato invece utilizzato un modulo GSM/GPRS, capace di interagire con il server per mezzo di una connessione web effettuata attraverso la rete telefonica mobile. Completano il sistema, uno schermo LCD per la visualizzazione diretta dei dati, un pack di batterie agli ioni di litio che fornisce l'alimentazione e un real time clock che provvede a coordinarne il funzionamento.

Le attività svolte dalla stazione sono essenzialmente di due tipi: acquisizione e trasmissione delle misure. La fase di sensing viene effettuata ogni ora ed è seguita dall'invio dei dati al server; una volta ultimati questi compiti, il sistema viene messo a dormire in modo da preservarne la carica elettrica. Questo processo dura in media circa novanta secondi e viene ripetuto ogni sessanta minuti. Durante il periodo di 'sleep' sarà possibile visualizzare sul display i dati inerenti all'ultima misura a disposizione.

Un'altra funzione permette di ricevere l'ultima lettura prodotta direttamente sul proprio telefono cellulare tramite un servizio di SMS; inviando infatti un messaggio alla stazione contenente una parola d'ordine, essa sarà in grado di processarlo e di rispondere al mittente con un secondo messaggio contenente i dati richiesti.

L'opportunità di affiancare al dispositivo un certo numero di moduli aggiuntivo, al fine di espanderne le funzionalità, punta a favorire lo sviluppo futuro della piattaforma ed ad ampliarne gli orizzonti.

La gestione dei dati inviati periodicamente dalle stazioni viene eseguita attraverso una infrastruttura informatica situata sul dominio *floweather.com*, che si occupa di immagazzinare, aggregare ed infine presentare le misure agli utenti. Per ciascuna stazione è infatti creato uno specifico file XML all'interno del quale sono memorizzate tutte le trasmissioni effettuate. La visualizzazione dei dati avviene invece per mezzo della pagina *MyFloweather*, una sezione personale all'interno dello spazio web dove ogni singolo utente è in grado di consultare in maniera immediata le misure trasmesse dal proprio dispositivo. Dotata di un'interfaccia grafica semplice e intuitiva, raccoglie una serie di grafici che riassumono l'andamento delle grandezze in questione, mostrando anche l'evoluzione dei valori massimi minimi e medi nel tempo.

Due ulteriori funzioni permettono infine di trasportare le proprie misure all'interno della pagina personale del social network Twitter, così da condividerle con gli altri utenti del popolare servizio di microblogging e di scaricare la totalità dei propri dati che è resa disponibile nello speciale formato EPW per realizzare simulazioni energetiche con software di progettazione come EnergyPlus o Climate Consultant.



Figura 2: Logo del progetto

# Capitolo 1

## Un dispositivo per misurazioni distribuite a metà tra una rete di sensori e un sistema embedded

### 1.1 Wireless Sensor Networks

#### 1.1.1 Definizione

Con il termine *Wireless Sensor Network* (o *WSN*) si indica una determinata tipologia di rete che, caratterizzata da una architettura distribuita, è realizzata da un insieme di dispositivi elettronici autonomi in grado di prelevare dati dall'ambiente circostante e di comunicare tra loro [1].

I recenti progressi tecnologici nei sistemi microelettromeccanici (*MEMS*), nelle comunicazioni wireless e nell'elettronica digitale hanno permesso lo sviluppo di piccoli apparecchi a bassa potenza, dai costi contenuti, multifunzionali e capaci di interagire fra loro tramite tecnologia wireless a raggio limitato. Questi dispositivi, chiamati comunemente sensori, sensor node in inglese o *mote* principalmente nell'America settentrionale, sono formati da componenti in grado di rilevare grandezze fisiche (temperatura, luminosità, accelerazione), elaborare dati e scambiarsi informazioni. Secondo la norma tecnica UNI4546 'Misure e misurazioni, Termini e definizioni fondamentali' un sensore è definito come un particolare trasduttore che si trova in diretta interazione con il sistema misurato.

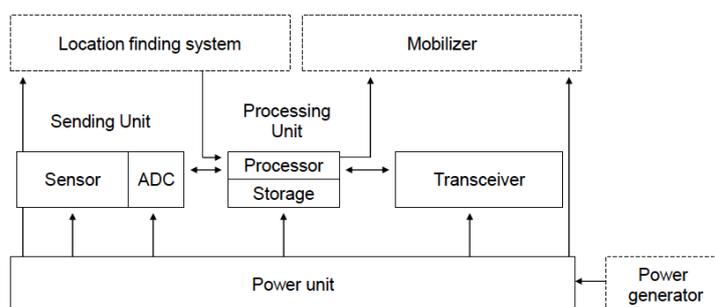


Figura 1.1: Schema di un generico nodo sensore

### 1.1.2 Caratteristiche

Una rete di sensori, o *sensor network*, è costituita da un insieme di nodi posizionati in prossimità oppure all'interno del fenomeno da osservare. Questi dispositivi sono prodotti e distribuiti in massa, hanno un costo di produzione trascurabile e sono caratterizzati da dimensioni e pesi molto ridotti. Ogni sensore ha una riserva d'energia limitata e non rinnovabile e, una volta messo in opera, deve essere in grado di lavorare autonomamente; per questo motivo tali dispositivi devono mantenere costantemente i consumi molto bassi, in modo da avere un maggior ciclo di vita. Per ottenere la maggior quantità possibile di dati occorre effettuare una massiccia distribuzione di sensori (nell'ordine delle migliaia o decine di migliaia) in modo da avere un'alta densità (fino a 20 nodi/m<sup>3</sup>) e far sì che i nodi siano tutti vicini tra loro, condizione necessaria affinché possano comunicare.

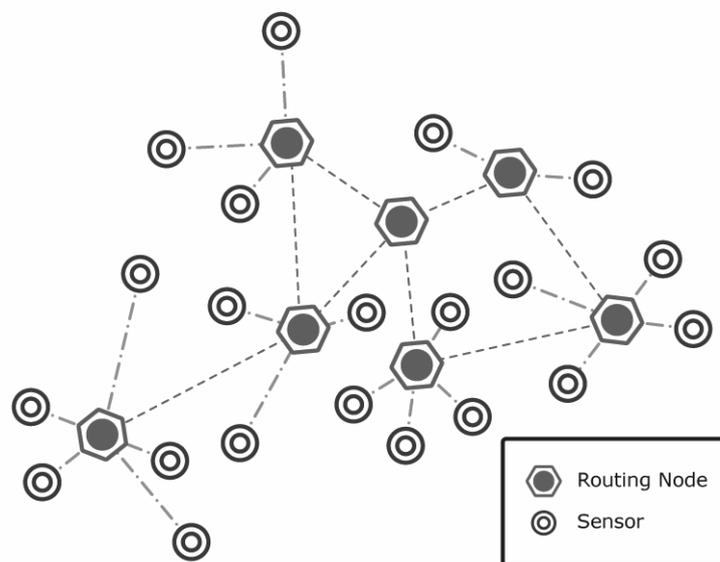


Figura 1.2: Schema di una generica rete di sensori

I nodi all'interno di una rete hanno la possibilità di collaborare tra loro dal momento che sono provvisti di un processore on-board; grazie a quest'ultimo, ciascuno di essi, invece di inviare dati grezzi ai nodi responsabili della raccolta dei dati, può effettuare delle semplici elaborazioni e trasmettere solo i dati richiesti e già elaborati.

La comunicazione, realizzata tramite tecnologia wireless a corto raggio, è solitamente di tipo asimmetrico, in quanto i nodi comuni inviano le informazioni raccolte ad uno o più nodi speciali della rete detti *sink*, i quali hanno lo scopo di mettere insieme i dati e trasmetterli tipicamente ad un server o ad una stazione di monitoraggio. Una comunicazione può avvenire autonomamente da parte del nodo nel momento in cui si verifica un dato evento o può venire indotta dal sink tramite l'invio di una *query* verso i nodi interessati.

I nodi sono sparpagliati in un'area chiamata *area di sensing*. Ciascun dispositivo, all'interno di questo spazio, ha la capacità di accumulare e di instradare i dati fino al sink, e quindi all'utente

finale. Il nodo sink è provvisto di un'antenna capace di illuminare tutto il dominio occupato dai sensori, che coincide con l'area di sensing. La posizione dei nodi all'interno della rete non deve essere predeterminata, in modo da utilizzare tale tecnologia in posti difficilmente accessibili o in operazioni di soccorso in luoghi disastriati per i quali è necessaria una disposizione random dei nodi.

Ciò significa che gli algoritmi e i protocolli utilizzati nelle reti di sensori devono possedere capacità auto organizzative. I sensori, conoscendo le caratteristiche di trasmissione del sink, e sfruttando il suo segnale di illuminazione (che possono sentire o no, a seconda dei casi), hanno la possibilità di effettuare un autoapprendimento di posizione, permettendo in tal modo la distribuzione casuale degli stessi nella rete. La capacità di autoapprendimento dei sensori è perciò una delle caratteristiche più importanti di questi ultimi. Tuttavia, a causa della scarsa quantità di energia di cui è dotato un sensore, occorre cercare di ottimizzare al meglio gli algoritmi che permettono allo stesso di conoscere la propria posizione, abbassando il suo tempo di apprendimento e puntando a minimizzare il tempo in cui un sensore deve stare sveglio ed il numero di volte in cui esso si deve svegliare.

Le reti di sensori possono essere utilizzate in numerose applicazioni, la cui realizzazione, per la maggior parte dei casi, richiede l'impiego di tecniche adottate anche nelle reti wireless ad hoc. Sfortunatamente però, molti degli algoritmi usati nelle reti ad hoc non sono compatibili con i requisiti di questo tipo di reti.

I principali motivi derivano dal fatto che:

- il numero di nodi che compongono una rete di sensori può essere di alcuni ordini di grandezza superiore rispetto al numero di nodi in una rete ad hoc;
- i nodi sono disposti con un'alta densità;
- i nodi sono soggetti a guasti;
- la topologia di una rete di sensori può cambiare frequentemente a causa di guasti ai nodi o della loro mobilità;
- i nodi utilizzano un paradigma di comunicazione broadcast mentre la maggior parte delle reti ad hoc sono basate su una comunicazione di tipo punto-punto;
- i nodi sono limitati rispetto ad alimentazione, capacità di calcolo e memoria;
- i nodi solitamente non possiedono un identificatore globale (come l'indirizzo IP nei computer);
- i nodi necessitano di una stretta integrazione con le attività di rilevamento.

Per questo motivo, questa tipologia di rete richiede algoritmi pensati e realizzati in maniera specifica per gestire la comunicazione e l'instradamento dei dati.

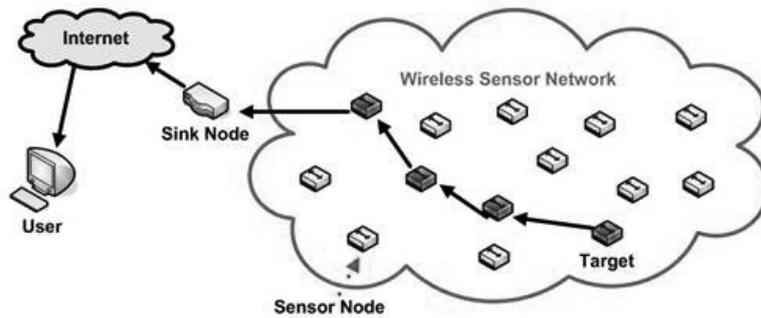


Figura 1.3: Trasmissione dell'informazione dal nodo attraverso il sink fino all'utente finale

Dal canto loro le reti di sensori possono migliorare in modo significativo la qualità delle informazioni, garantendo un'elevata fedeltà, fornendo informazioni in tempo reale anche in ambienti ostili e riducendo i costi di trasmissione delle informazioni stesse.

Lo scopo fondamentale di questo tipo di reti è di produrre in un periodo esteso di tempo un'informazione globale significativa, ottenuta a partire da una serie di dati locali provenienti dai singoli sensori. È importante notare che la rete deve essere realizzata in modo da garantire l'integrità dei rilevamenti per un lasso di tempo che sia il più lungo possibile, allo scopo di ottenere informazioni accurate anche in caso di attacco da parte di organi esterni o di cedimenti hardware.

Il fatto che un singolo sensore sia dotato di una piccola quantità di energia non deve impedirgli di inviare le informazioni elaborate, che verranno successivamente raccolte e unite a quelle provenienti dagli altri sensori. Esistono diverse tecniche che possono essere usate per connettere la rete con l'esterno, in particolare per trasmettere le informazioni che essa raccoglie; in genere, i nodi sink all'interno della rete, che hanno il compito di convogliare a sé tutte le informazioni richieste, agiscono anche come gateway provvedendo alla trasmissione dei dati verso i centri di elaborazione.

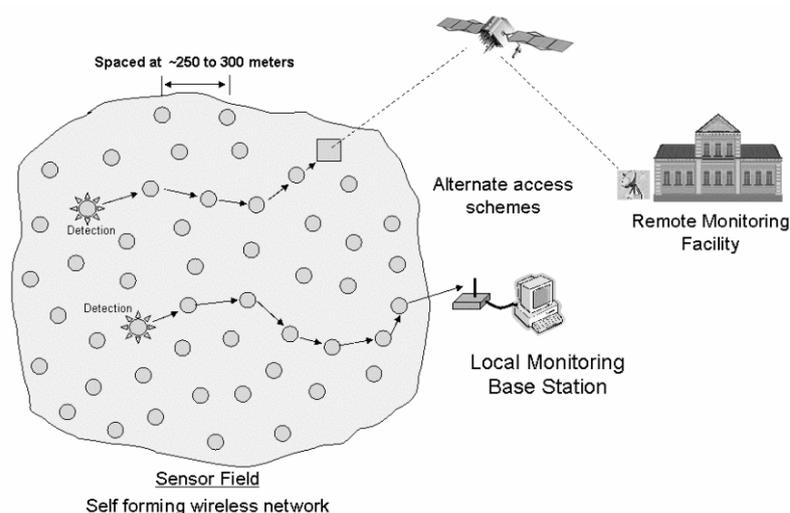


Figura 1.4: Modalità di monitoraggio di un'area attraverso l'utilizzo di sensori

### 1.1.3 Applicazioni

Come accennato in precedenza, le particolari caratteristiche delle WSN permettono di avere un alto grado di applicabilità in un vastissimo numero di ambiti.

Le applicazioni in cui è possibile usare reti di sensori, solitamente, sono classificabili in militari, ambientali, sanitarie, casalinghe e commerciali [2].

#### 1.1.3.1 Applicazioni militari

La facilità e rapidità di distribuzione, l'auto-organizzazione e la tolleranza ai guasti rendono la rete di sensori una soluzione ottimale per applicazioni militari.

Poiché queste reti sono basate su una disposizione densa di nodi monouso ed a basso costo, la distruzione di alcuni nodi da parte del nemico non danneggia le operazioni militari come al contrario potrebbe accadere con l'impiego di dispositivi tradizionali. Le possibili applicazioni vanno dal monitoraggio di forze alleate, equipaggiamenti e munizioni alla sorveglianza del campo di battaglia.

Sempre nel campo militare, è possibile usare una rete di sensori per effettuare il riconoscimento di nemici, la stima dei danni di una battaglia oppure l'identificazione del tipo di attacco (nucleare, biologico o chimico).

#### 1.1.3.2 Applicazioni ambientali

In tale ambito, le reti di sensori sono comunemente usate per alcune applicazioni che riguardano, ad esempio, il monitoraggio del movimento degli uccelli, il controllo della flora in una data area a rischio o la raccolta dei dati climatici per studi meteorologici. È possibile, inoltre, effettuare un controllo costante delle foreste, al fine di rilevare prontamente eventuali incendi.

Negli Stati Uniti è stato messo a punto un sistema per la previsione e rilevazione di inondazioni. Tale sistema si chiama *Alert System* ed è formato da molti sensori in grado di rilevare la quantità di pioggia caduta, il livello dell'acqua di un fiume e le condizioni climatiche, come ad esempio temperatura o umidità [3].

Le reti di sensori possono essere utilizzate anche nell'agricoltura di precisione. Tra gli obiettivi possibili da raggiungere vi è quello del monitoraggio dei livelli di pesticidi nell'acqua, di erosione del terreno e di inquinamento dell'aria; il tutto effettuato in tempo reale.

Altra applicazione è quella del monitoraggio dell'habitat di alcune specie animali, e lo studio del loro comportamento.

Sempre nel settore ambientale, questa tipologia di reti può risultare utile per studiare gli spostamenti e il dinamismo all'interno dei ghiacciai. In tal caso i sensori vengono distribuiti all'interno del ghiaccio a profondità differenti e sono capaci di rilevare temperatura e pressione, comunicando con una stazione base posizionata in cima al ghiacciaio che provvederà, a sua volta, al trasferimento dei dati.

### 1.1.3.3 Applicazioni medico-sanitarie

I possibili utilizzi delle reti di sensori in questo campo sono rivolti a fornire un'interfaccia per le persone affette da handicap, al monitoraggio dei dati fisiologici del paziente, all'amministrazione ospedaliera, sia essa relativa alla gestione dei farmaci che al controllo di medici e pazienti all'interno della struttura.

E' inoltre possibile applicare direttamente sul paziente dei piccoli sensori che sono in grado di fornire alcuni dati, come la frequenza cardiaca o il tasso di saturazione dell'ossigeno nell'organismo.

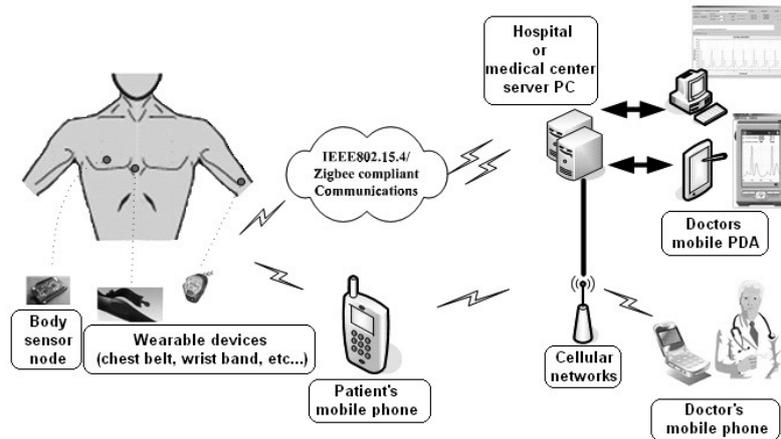


Figura 1.5: Esempio di trasmissione di informazioni dal paziente al medico per mezzo di WSN

### 1.1.3.4 Applicazioni domestiche

L'utilizzo di una rete in ambito casalingo è strettamente legato all'ambito della domotica, la scienza che si occupa dello studio delle tecnologie impiegate a migliorare la qualità della vita nella casa. Essa ha lo scopo di trovare strumenti e strategie per migliorare la sicurezza, risparmiare energia e ridurre i costi di gestione. Un esempio di utilizzo consiste nel dotare tutti gli elettrodomestici della casa di sensori, i quali, controllando questi apparecchi, possono interagire direttamente l'uno con l'altro o con reti esterne, tramite l'utilizzo di internet, in modo da poter essere gestiti anche da distanze remote. L'ambiente domestico viene ad assumere così le stesse caratteristiche di una struttura centralizzata, connessa per mezzo di una rete e in grado di mettere in comunicazione tra loro tutti gli strumenti di cui l'ambiente è composto.

### 1.1.3.5 Applicazioni commerciali

In questo ambito alcuni esempi possono essere forniti da applicazioni mirate a prevenire eventuali crolli di ponti o grattacieli attraverso delle misurazioni capaci di riscontrarne difetti strutturali. Un altro ambito applicativo interessante è quello relativo al rilevamento della posizione e del movimento di veicoli o *car tracking* in cui, tramite una rete di sensori, è possibile monitorare la posizione di un'auto in movimento, ovvero il traffico su strade particolarmente transitate.

### 1.1.4 Architettura di comunicazione

La tipica architettura utilizzata nelle reti di sensori è quella stratificata che fa riferimento allo standard dello *stack OSI* e che prevede i livelli fisico, data link, di rete, di trasporto e di applicazione.

Tuttavia, in questa tipologia di reti ogni livello è segmentato attraverso tre diversi piani, ciascuno dei quali è responsabile della gestione di alcuni aspetti di rete.

I tre sottolivelli prendono il nome di:

1. *power management plane*, il quale gestisce i consumi energetici del sensore, facendo in modo, quando l'energia restante nel nodo non è sufficientemente alta, di escluderlo dal processo di routing, utilizzando la potenza rimanente per la sola fase di sensing;
2. *mobility management plane*, il quale rileva e memorizza le coordinate dei sensori in modo da conoscere sempre la posizione degli stessi all'interno dell'ambiente e quella dei nodi vicini, bilanciando così la potenza di trasmissione;
3. *task management plane*, il quale è responsabile delle fasi di raccolta ed elaborazione dei dati, che vengono programmate in base agli obiettivi di sensing stabiliti.

Gli strati che contraddistinguono maggiormente una WSN, differenziandola così dalle altre, e che sono stati progettati in maniera specifica per soddisfarne le necessità, sono quelli di applicazione, di rete e di accesso al mezzo.

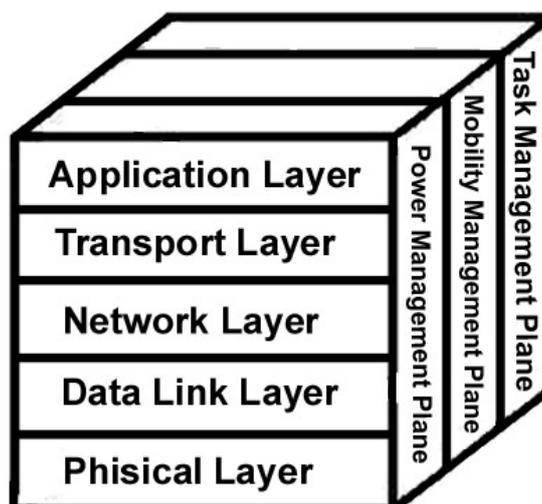


Figura 1.6: Stack protocollare di una rete di sensori

#### 1.1.4.1 Livello di applicazione

La gestione degli applicativi presenti nei sensori della rete è effettuata, da parte degli amministratori di sistema, attraverso il protocollo *SMP* (*Sensor Management Protocol*), che permette di effettuare diverse operazioni quali:

- accensione e spegnimento dei nodi;
- spostamento dei nodi;
- riconfigurazione della rete;
- distribuzione di chiavi, autenticazione e sicurezza;
- sincronizzazione temporale dei nodi;
- scambio di dati relativo ad algoritmi per la localizzazione.

Gli amministratori possono richiedere, in qualsiasi momento, dati alla rete attraverso un procedimento basato su query, che prende il nome di *SQTL* (*Simple Query and Tasking Language*), simile al più noto *SQL*. Un utente specifica al nodo sink le informazioni di cui necessita; esso genera quindi un'interrogazione che viene poi immessa nella rete. Una o più risposte sono rinviate indietro al sink, il quale riunisce ed elabora queste risposte prima di fornire il risultato finale al richiedente.

Le reti di sensori sono quindi modellate come dei database distribuiti in cui il flusso delle informazioni dipende dal meccanismo di elaborazione interno al database.

Oltre al *SMP* esistono altri tipi di protocolli che sono comunemente utilizzati in queste reti.

I più famosi sono:

- *TADAP* - *Task Advertisement and Data Advertisement Protocol*;
- *SQDDP* - *Sensor Query and Data Dissemination Protocol*.

#### 1.1.4.2 Livello di rete

Il livello di rete si occupa di instradare i dati attraverso il percorso più efficiente dal punto di vista energetico, in base alla potenza disponibile ai nodi o all'energia necessaria per la trasmissione. Come affermato in precedenza, a causa delle caratteristiche peculiari, le reti di sensori non possono utilizzare algoritmi di routing convenzionali, ma necessitano protocolli specifici.

La maggior parte dei protocolli impiegati nelle WSN possono essere raggruppati in tre grandi famiglie:

1. *Data Centric Protocols*, i quali sono progettati per far fronte alla difficoltà di assegnare un identificativo a ciascun nodo della rete, specie se in gran numero e disposti casualmente nell'ambiente; gestiscono il routing basandosi sulla descrizione dei dati, piuttosto che su quella dei nodi. Vengono utilizzati quando gli utenti sono maggiormente interessati ad effettuare un'interrogazione per conoscere un attributo inerente al fenomeno osservato, piuttosto che il valore ad un dato nodo. Ne sono alcuni esempi gli algoritmi *SPIN*, *Direct Diffusion* o di *flooding*.
2. *Hierarchical Protocols*, i quali sono stati proposti per cercare di risolvere i problemi riguardanti la scalabilità e i consumi energetici della rete. Per far ciò i nodi si aggregano in gruppi o *cluster* e inviano le proprie misurazioni ad un unico nodo, detto *cluster-head*, il quale ha lo scopo di raccogliere, aggregare e fondere insieme i dati in modo da risparmiare energia. Tutti i *cluster-head* all'interno della rete possono aggregarsi a loro volta, formando uno strato superiore, così da inviare tutti i dati al sink tramite un unico *cluster-head*. Il protocollo più famoso di questa famiglia è denominato *Low-Energy Adaptive Clustering Hierarchy (LEACH)*.
3. *Location Based Protocols*, i quali permettono ai protocolli di routing, se la posizione dei nodi all'interno della rete è nota, di usare questa informazione al fine di ridurre la latenza e i consumi nella rete stessa. Un esempio è il protocollo *Minimum Energy Communication Network (MECN)*, che rappresenta la rete come un grafo connesso, a partire dal quale calcola un sottografo che connette tutti i nodi al costo energetico minimo.

### 1.1.4.3 Livello MAC

Lo strato di accesso al mezzo, o *Medium Access Control*, assolve due particolari funzioni.

Per prima cosa si occupa di creare un'infrastruttura di rete comune, in modo offrire un collegamento alle centinaia di nodi che si trovano dispersi nell'ambiente. Esso permette di stabilire i link di comunicazione al fine di abilitare la trasmissione dei dati. Questo provvede a formare l'infrastruttura di base necessaria per le comunicazioni wireless, oltre a fornire ai nodi la capacità di riorganizzarsi.

In secondo luogo, come conseguenza, consente un'efficiente condivisione delle risorse tra nodi della rete.

Come già accennato in merito agli algoritmi impiegati, a causa delle particolari caratteristiche di questo tipo di reti, per difficoltà principalmente legate alla durata limitata della batteria e alla corrispondente necessità di ridurre al meglio i consumi, non è possibile utilizzare i protocolli MAC esistenti. Ad esempio nelle reti cellulari, dove le *Base Station* formano una rete dorsale cablata e sono alimentate in maniera costante, i nodi, rappresentati dai telefoni cellulari, dispongono di una batteria che può essere ricaricata dall'utente in qualsiasi momento. Estremamente diverso appare il caso delle reti wireless di sensori, le quali non prevedono la presenza di un agente di

controllo centrale come una *BS* e in cui l'efficienza energetica influenza direttamente il ciclo vitale funzionamento dell'infrastruttura, diventando un fattore di prima importanza.

L'insieme di queste esigenze ha portato nel tempo allo sviluppo di protocolli MAC energeticamente efficienti, che consentissero di ridurre i consumi mantenendo, ad esempio, la radio spenta per gran parte del tempo.

Una soluzione possibile è quella del *Low Power Listening (LPL)*, che riduce il tempo di ascolto tramite l'introduzione di un ciclo di lavoro che opera a livello fisico. L'idea base è quella di spostare i costi dal ricevitore al trasmettitore, aumentando così la lunghezza del preambolo che indica la necessità di trasmettere un'informazione; questa tecnica consente al ricevitore di attivarsi solo periodicamente all'ascolto del canale per rilevare un preambolo: in caso positivo il nodo rimane acceso, altrimenti torna a dormire. Ciò produce un aumento della latenza ed una riduzione del throughput.

Un'altro protocollo con simili caratteristiche è il *Self Organizing Mac*, o *S-MAC*, il quale si occupa di gestire gli overhead causati da collisioni o dall'overhearing. Il concetto fondamentale è di dividere il tempo in più frame, ciascuno dei quali è composto da due parti, una attiva ed una di 'sleep'. Poiché il periodo di inattività risulta di gran lunga superiore a quello di funzionamento, i messaggi da trasmettere vengono inizialmente salvati in memoria in questa fase per poi essere inviati quando il nodo è attivo. L'intervallo di attività ha solitamente una durata di circa 300ms, mentre il frame può essere impostato di qualsiasi lunghezza.

### 1.1.5 Progetto di rete

La progettazione di una rete di sensori è influenzata da diversi fattori, i quali, non solo risultano necessari per la progettazione stessa, ma influenzano a loro volta la scelta degli algoritmi utilizzati all'interno della rete.

Questi fattori sono:

- tolleranza ai guasti;
- scalabilità del sistema;
- costi di produzione;
- ambiente operativo;
- topologia della rete;
- vincoli hardware;
- mezzo di trasmissione.

### 1.1.5.1 Tolleranza ai guasti

Nel progettare un qualsiasi tipo di rete, comprese le reti di sensori, occorre sempre considerare la possibilità che alcuni nodi della stessa siano affetti da malfunzionamenti o guasti, le cui cause possono essere danni fisici, interferenze o batterie scariche. La tolleranza ai guasti è la capacità di far funzionare una rete di sensori anche in caso di malfunzionamento da parte di alcuni nodi.

La tolleranza ai guasti (o affidabilità) di un nodo  $k$  è modellata tramite una distribuzione di Poisson ed è vista come la probabilità di non avere malfunzionamenti nell'intervallo  $(0, t)$ :

$$R_k(t) = e^{-\lambda kt}$$

Dove il fattore  $\lambda k$  è il tasso di fallimento del nodo  $k$ -esimo.

I protocolli e gli algoritmi possono essere progettati in modo da garantire il livello di tolleranza ai guasti richiesto dalla rete. Tale livello dipende fortemente dall'applicazione in cui viene utilizzata la rete di sensori (uso militare, domestico, commerciale, etc.).

### 1.1.5.2 Scalabilità

Con il termine scalabilità si intende la capacità del sistema di funzionare sia all'aumentare che al diminuire del numero di nodi all'interno della rete (il cui totale può variare da qualche unità fino ad oltre un milione). Uno dei metodi per poter ottenere un sistema scalabile è quello di riuscire a sfruttare la natura densa delle reti di sensori. La densità di nodi in una rete di sensori risulta *application dependent*, ossia strettamente legata al tipo di applicazione per cui la rete è realizzata, e, in una regione di diametro minore di dieci metri, può arrivare a superare il centinaio di unità.

La densità, definita come il numero di nodi presenti all'interno del raggio di trasmissione di un dato nodo, può essere calcolata nel seguente modo:

$$\mu(r) = \frac{(N\pi r^2)}{2}$$

dove  $N$  è il numero di nodi sparpagliati in una regione di area  $A$ , ed  $r$  è il raggio di trasmissione di un nodo.

### 1.1.5.3 Costi

Poiché una rete di sensori è formata da un grande numero di nodi, il costo di un singolo elemento è molto importante e risulta critico, in quanto, se il costo totale della rete è superiore rispetto a quello prodotto dall'utilizzo di tecnologie di sensing tradizionali, allora l'uso di una WSN non è giustificabile.

Il costo di un singolo sensore dovrebbe perciò essere abbastanza basso (tipicamente minore di un euro). Questo non è un obiettivo molto facile da raggiungere poiché, al di là dei componenti utilizzati per il sensing, i quali sono generalmente economici, è necessario che i nodi siano dotati

anche di un modulo per la comunicazione (che a seconda delle esigenze può essere basato sugli standard *WiFi*, *GSM*, *ZigBee* o *Bluetooth*), il cui prezzo varia dalle decine alle centinaia di euro. Un nodo è poi di norma provvisto di altre unità come un processore o un campionatore; inoltre, in alcune applicazioni può essere necessario equipaggiare un nodo con un sistema di posizionamento globale (*GPS*). Tutti questi fattori portano naturalmente ad un incremento del costo finale del sensore.

#### 1.1.5.4 Ambiente operativo

Come sottolineato in precedenza, i sensori sono in genere disposti in prossimità o addirittura all'interno del fenomeno da osservare; perciò, spesso, si trovano a lavorare in zone geografiche remote (all'interno di un macchinario, in fondo all'oceano, sulla superficie dell'oceano durante un tornado, in una zona biologicamente o chimicamente contaminata, in un campo di battaglia, etc.) e senza la supervisione dell'uomo.

Tutto ciò dà un'idea delle condizioni sotto le quali i sensori devono essere capaci di funzionare, dovendo sopportare ad esempio alte pressioni qualora dovessero lavorare in fondo all'oceano, alte o basse temperature, esposizione a fattori naturali o artificiali che, col tempo, possono danneggiarli.

#### 1.1.5.5 Topologia di rete

I nodi sono disposti l'uno accanto all'altro e spesso anche con un'alta densità. Questo richiede un'attenta attività per il mantenimento della topologia.

Il mantenimento e il cambiamento della topologia può essere diviso in tre fasi:

1. *pre-deployment e deployment phase*: in questa prima fase i sensori vengono disposti nell'ambiente da osservare; questa operazione può essere effettuata in maniera casuale (ad esempio gettando i sensori da un aereo) o puntuale, collocandoli uno ad uno in posizioni specifiche;
2. *post-deployment phase*: una volta effettuato il posizionamento e realizzata una topologia di partenza, si affrontano i successivi cambiamenti topologici della rete, che sono dovuti a molteplici situazioni, dal cambiamento della posizione dei nodi alla variazione della raggiungibilità di un nodo, dall'energia disponibile alla presenza di malfunzionamenti;
3. *re-deployment of additional nodes phase*: infine, i nodi addizionali possono essere ridisposti in qualsiasi momento per rimpiazzare quelli mal funzionanti o a causa della variabilità dinamica degli obiettivi di misurazione. Naturalmente l'aggiunta di nuovi nodi comporta la necessità di riorganizzare la rete. L'alta frequenza di cambiamenti topologici, uniti al vincolo stringente del risparmio energetico, richiedono perciò protocolli di routing molto particolari.

### 1.1.5.6 Vincoli hardware

Un sensore è composto da quattro unità base:

1. *unità di sensing*: di solito è composta da due sotto unità, sensore e convertitore analogico digitale;
2. *unità computazionale*: è generalmente associata ad una piccola unità di immagazzinamento dati e gestisce le procedure che permettono la collaborazione del nodo con gli altri dispositivi della rete per portare a termine il task assegnato;
3. *unità transceiver*: è l'unità che connette il nodo alla rete, può essere costituita da un apparecchio ottico o a radio frequenza (*RF*);
4. *unità energetica*: è forse il componente più importante di un nodo sensore, a volte può essere supportata da una unità per il recupero di energia, come ad esempio una fonte solare.

### 1.1.5.7 Mezzo trasmissivo

In una rete di sensori multihop i nodi possono interagire tra loro tramite un mezzo di comunicazione wireless; è quindi possibile utilizzare onde radio. Una opportunità è quella di usare le bande *ISM* (*Industrial, Scientific and Medical*), un gruppo predefinito di bande che in molti paesi sono utilizzabili liberamente. La maggior parte dei sensori attualmente in commercio fanno infatti uso di un circuito RF.

Un altro possibile modo per far comunicare i nodi è tramite la radiazione infrarossa (*IR*). La comunicazione con gli infrarossi non richiede permessi o licenze; inoltre i *transceiver* basati su questa tecnica sono economici e facili da costruire. Tuttavia, il problema maggiore che si ha nell'uso degli infrarossi consiste nella necessità di avere un interfacciamento diretto tra il trasmettitore ed il ricevitore. Questo rende tale tecnologia di difficile utilizzo nelle reti di sensori dove solitamente i nodi sono disposti in modo casuale.

La scelta del mezzo trasmissivo, d'altra parte, è spesso imposta dall'applicazione. Ad esempio, le applicazioni marine impongono di usare l'acqua come mezzo trasmissivo; in questo caso sarebbe quindi necessario utilizzare segnali in grado di propagarsi nell'acqua.

## 1.2 Sistemi embedded

### 1.2.1 Definizione

Con il termine *embedded system* (tradotto in italiano con 'sistema incorporato') si identificano genericamente tutti quei sistemi elettronici che, dotati di un microprocessore, sono progettati appositamente per una determinata applicazione (*special purpose*) e non sono riprogrammabili dall'utente per altri scopi; sono inoltre spesso supportati da una piattaforma hardware su misura, integrati all'interno del sistema da controllare ed in grado di gestire tutte o parte delle funzionalità [4].

In tale area si collocano sistemi di diverse tipologie e dimensioni, in base al tipo di microprocessore, al sistema operativo e alla complessità del software, che può variare da poche centinaia di byte a parecchi megabyte di codice. Appartengono a questa categoria di sistemi microelettronici di elaborazione anche i microcontrollori.

Contrariamente ai dispositivi elettronici di tipo *general purpose*, che non sono dedicati ad un solo possibile utilizzo (come ad esempio i personal computer), un sistema embedded svolge dei compiti noti già durante la fase di sviluppo, che vengono eseguiti grazie ad una combinazione hardware/software specificamente studiata per la determinata applicazione. Grazie a ciò l'hardware può essere ridotto ai minimi termini in modo da contenere lo spazio occupato, i consumi ed i costi di fabbricazione. Inoltre, l'esecuzione del software avviene spesso in tempo reale per permettere un controllo deterministico dei tempi di esecuzione.

Un sistema embedded si differenzia quindi da un comune sistema a microprocessore come un PC, poiché:

- è studiato per svolgere un particolare compito;
- ha risorse e dispositivi strettamente necessari per assolvere alla sua funzione;
- possiede un hardware dedicato;
- lavora a frequenze di CPU inferiori;
- deve mantenere bassi i consumi energetici;
- gestisce poca memoria.

Alcuni esempi diffusi di sistema embedded sono le centraline elettroniche installate a bordo degli autoveicoli per il controllo del motore e dell'ABS, le macchine fotocopiatrici, i telefoni cellulari o vari tipi di elettrodomestici quali forni a microonde, lavatrici, apparecchi televisivi o lettori DVD.

Tuttavia, ad oggi, è difficile pensare ad un dispositivo elettronico che non includa al suo interno almeno un sistema di tipo embedded.

## 1.2.2 Caratteristiche

I sistemi embedded sono sistemi di calcolo, nel significato più generale del termine. Questa definizione include, infatti, ogni tipo di computer, eccetto quelli progettati per essere di utilità generica (general purpose). Gli esempi di sistemi embedded spaziano dai lettori portatili di musica ai controlli in tempo reale di sistemi fisici come lo Space Shuttle.

Un generico sistema embedded possiede solitamente quattro caratteristiche principali:

1. specializzazione e ottimizzazione dell'applicazione svolta;
2. inclusione di hardware e software dedicati per:
  - interagire con il mondo esterno,
  - svolgere alcune funzioni elaborative con diverse velocità;
3. necessità di rispettare vincoli di prestazioni minime, come operare in real-time;
4. implementazione sotto forma di sistemi distribuiti o componenti di sistemi distribuiti.

### 1.2.2.1 Specializzazione

La maggior parte dei sistemi embedded è progettata per eseguire ripetutamente una sola o al massimo poche specifiche applicazioni a costo contenuto. Tali applicazioni sono note a priori, prima che il processo di progettazione inizi. Spesso è opportuno garantire al sistema la flessibilità necessaria per i futuri aggiornamenti o per un eventuale riutilizzo del componente; normalmente questo scopo è raggiunto rendendo il sistema riprogrammabile.

Inoltre, gran parte di questi sistemi ha capacità native di auto-verifica delle proprie funzionalità. In genere, in caso di procedure in *loop* o in situazioni di *deadlock*, intervengono dei meccanismi di protezione chiamati *watchdog*. Ci sono diversi tipi di verifiche, divise in base alla funzione (alimentazione, comunicazione, consumo) o al componente (CPU, RAM, periferiche) controllati.

### 1.2.2.2 Hardware dedicato

I sistemi embedded rispettano i vincoli di prestazione grazie a un insieme di hardware e software appositamente progettati; per la maggior parte di essi le prestazioni richieste possono essere soddisfatte, infatti, con una combinazione di hardware dedicato e una quantità limitata di software ottimizzato. Spesso, quindi, gran parte dell'hardware di sistema deve sottostare a requisiti di prestazioni molto meno severi di quelli che, invece, deve rispettare l'hardware primario del sistema stesso. Questo permette all'architettura di tale sistema di essere semplificata rispetto a quella di un generico computer che deve eseguire le stesse operazioni. Il software scritto per molti sistemi embedded, in particolare quelli senza hard disk, è talvolta chiamato *firmware*.

L'interazione del sistema col mondo esterno, di per sé analogico, rende indispensabile per un sistema la capacità di raccogliere, campionare e trasmettere segnali. In aggiunta, i sistemi embedded possono essere fisicamente inaccessibili (come per le trivelle dei pozzi di petrolio oppure per i componenti lanciati nello spazio); pertanto i sistemi che li contengono devono essere capaci di resettarsi autonomamente in caso di perdita o corruzione dei dati.

### 1.2.2.3 Applicazioni real-time

Un sistema real-time persegue degli obiettivi a partire dai vincoli temporali forniti. A seconda delle applicazioni per cui è progettato si può operare una distinzione tra sistemi *hard real time*, nei quali i task devono terminare entro specifici limiti temporali, e sistemi *soft real time*, in cui i task possono oltrepassare i limiti senza per questo causare guasti o malfunzionamenti importanti. I sistemi embedded appartengono alla tipologia hard real-time.

### 1.2.2.4 Sistemi distribuiti

L'innovazione tecnologica ha portato negli anni un aumento sul mercato del numero dei sistemi embedded distribuiti, così come delle componenti fornite da vari produttori per facilitarne la progettazione.

Questi sistemi sono composti da più sottosistemi che cooperano nello svolgimento di un servizio comune. Ciò porta degli indubbi vantaggi, come la localizzazione dell'elaborazione (il lavoro viene svolto dove necessario), la specializzazione dell'elaborazione (il lavoro è fatto da chi lo sa fare meglio) e la ridondanza (possibilità di supplire a guasti parziali), ma anche alcuni svantaggi, quali la necessità di coordinamento e comunicazione fra i vari sottosistemi ed un globale aumento della complessità [5].

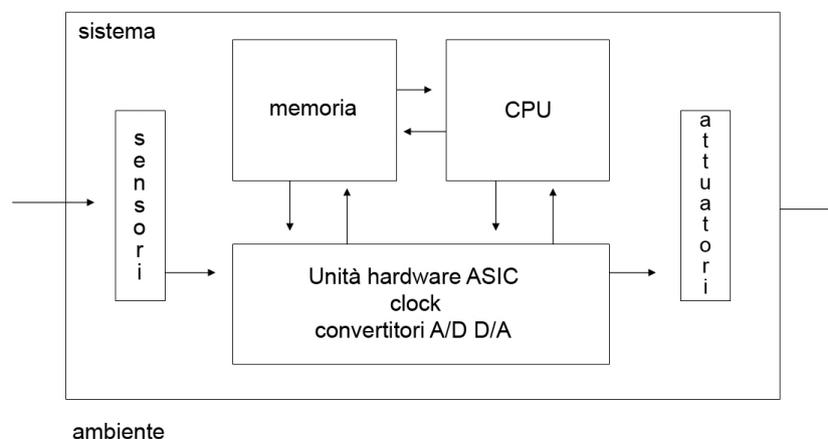


Figura 1.7: Schema di un generico sistema embedded

### 1.2.3 Applicazioni

Le applicazioni basate su sistemi embedded presentano diversi settori di impiego; tra i principali ricordiamo:

- acquisizione dati ed elaborazione;
- comunicazioni;
- sistemi di controllo digitale;
- robotica;
- interfacce;
- unità ausiliarie.

E' dunque embedded l'elettronica montata a bordo dell'auto o nel quadro di controllo di un moderno locomotore; è embedded il sistema che gestisce l'intelligenza di un telefonino, l'elettronica che sovrintende alle operazioni di un Bancomat o, ancora, il sistema che governa le complesse operazioni di controllo di un grosso impianto industriale che, ad esempio, estrude manufatti in alluminio.

Appare, quindi, facile da comprendere quanto sia importante l'elettronica di tipo incorporato nella vita di tutti i giorni: dalle operazioni di banale gestione quotidiana a quelle di grande importanza tecnica che riguardano i trasporti, la produzione, la sicurezza e via dicendo.

Tale tecnologia è in rapido sviluppo e si propone in tutte le direzioni: dal software all'hardware, dall'analogica al digitale, dall'automazione al consumer.

Alla base di questo sviluppo vi sono varie tecnologie cosiddette abilitanti, ossia quelle tecnologie che per la loro flessibilità e la loro potenzialità intrinseca promuovono la creazione di applicazioni sempre più integrate e innovative.

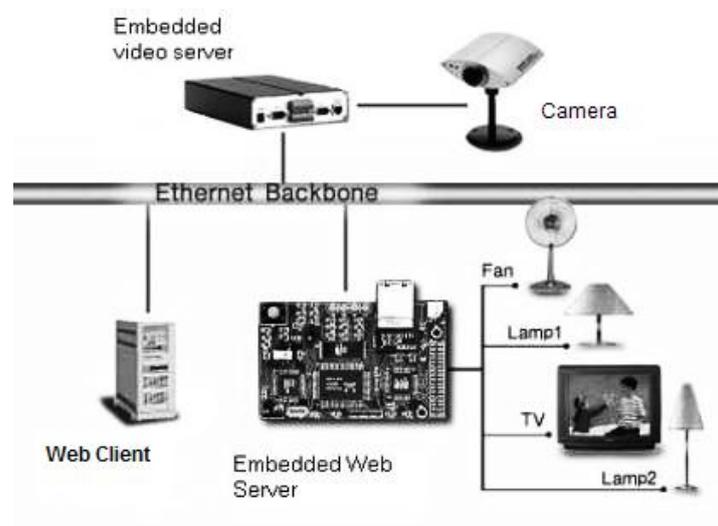


Figura 1.8: Esempio di applicazione embedded

## 1.2.4 Requisiti di progetto

Durante la fase di progetto di un qualsiasi sistema embedded occorre considerare aspetti di diverso genere, che influenzano alcune scelte progettuali e che dipendono dall'applicazione che si vuole realizzare.

La scelta delle CPU non è infatti l'unica che deve fare il progettista, il quale avrà il compito di determinare con che tipo di memoria equipaggiare il sistema, quante e quali periferiche aggiungere, se usare o meno un sistema operativo.

La piattaforma sulla quale può venire sviluppato un sistema embedded varia drasticamente a seconda della sua complessità, dei consumi energetici, del costo e del campo di utilizzo. Si passa dai microcontrollori più semplici, ad architetture più complesse basate su circuiti integrati sofisticati come i *System on a chip* (o *SoC*).

La presenza o meno di un sistema operativo completo su un sistema embedded varia, a sua volta, a seconda della sua complessità architetturale e del campo di utilizzo.

Su microcontroller semplici opererà ciclicamente un singolo programma di pochi byte senza alcuna sovrastruttura, su ambienti complessi possono al contrario trovare applicazione i medesimi sistemi operativi impiegati comunemente per scopi generali eventualmente personalizzati, sistemi più specializzati per gestire eventi in real-time o estremamente specializzati e non disponibili sul mercato (come i sistemi operativi dei cellulari GSM di prima generazione, per esempio, sviluppati abitualmente dal produttore degli apparati stessi).

Individuare i requisiti di un sistema rappresenta il punto di partenza fondamentale di qualsiasi attività progettuale. I requisiti di progetto imposti da un sistema embedded sono molteplici e possono spesso trovarsi in contrasto tra loro.

Tra i principali ricordiamo:

- requisiti funzionali;
- requisiti temporali;
- requisiti di affidabilità;
- consumi;
- prestazioni;
- costi.

### 1.2.4.1 Requisiti funzionali

Contengono la specifica della parte elaborativa del sistema, definendo i dati di ingresso e la loro provenienza, i dati di uscita e la loro destinazione, e le relazioni che esistono fra tra questi due tipi di dato, ovvero quali output devono essere prodotti dal sistema in funzione degli input.

### 1.2.4.2 Requisiti temporali

Possono derivare dalla necessità di raggiungere degli obiettivi prima di una determinata scadenza, o di eseguirli periodicamente. Altri fattori come la latenza nella risposta o nella rilevazione degli errori rappresentano parametri di notevole importanza.

### 1.2.4.3 Requisiti di affidabilità

L'affidabilità di un sistema può essere stimata a partire dai seguenti parametri:

- *MTTF - Mean Time To Failure*: è il tempo medio che intercorre tra due errori, più grande è l'intervallo maggiormente affidabile risulterà il sistema;
- *MTTR - Mean Time To Repair*: è il tempo medio necessario per riparare un errore, indica il grado di riparabilità del sistema ed è inversamente proporzionale alla sua affidabilità;
- $D = MTTF / (MTTF + MTTR)$ : rappresenta la disponibilità, ossia la percentuale di tempo in cui il sistema è funzionante.

### 1.2.4.4 Consumi

Il consumo energetico è uno dei requisiti fondamentali per un sistema embedded; esso influisce direttamente sul costo complessivo e sulla complessità dell'hardware, in particolar modo per quanto riguarda alimentatori, batterie, sistemi di raffreddamento. In sistemi alimentati da batteria è fondamentale per assicurare una maggiore autonomia e, di conseguenza, un periodo di vita maggiore del dispositivo.

Il consumo, infine, è direttamente collegato al problema della dissipazione del calore e del rumore, anch'essi da non sottovalutare.

### 1.2.4.5 Prestazioni

La velocità del sistema influenza diversi aspetti: con prestazioni migliori, ad esempio, si soddisfano più facilmente i requisiti temporali e si aumenta l'usabilità del sistema stesso, ma, allo stesso tempo, si sperimenta un aumento nei consumi del sistema. E' possibile tuttavia utilizzare strategie specifiche per ridurre gli aspetti negativi, come variare dinamicamente le prestazioni in modo da controllarne il consumo.

Un aumento delle prestazioni si riflette, per di più, quasi sempre in un aumento del costo finale del dispositivo.

### 1.2.4.6 Costi

Per sistemi prodotti in larga scala il costo finale è un aspetto fondamentale. Il costo è profondamente legato alle scelte di progetto; i principali fattori che lo influenzano sono:

- scelta dell'architettura (distribuita, centralizzata o ibrida);
- hardware (tipo di CPU, tipo e quantità di memoria, periferiche I/O necessarie);
- software (inclusi costi di progettazione e sviluppo);
- licenze e diritti per HW e SW (librerie, compilatori, ambienti di sviluppo e di testing);
- numero di pezzi prodotti.

Risulta difficile caratterizzare in maniera specifica i costi di un sistema embedded generico, anche se, soprattutto per sistemi che devono processare una grande quantità di dati, il progetto stesso assorbe la maggior parte dei costi.

Nel caso di sistemi che devono essere commercializzati in massa, come un lettore di musica portatile, ridurre i costi diventa una priorità. Sistemi di questo genere, spesso, sono infatti dotati di alcuni chip, una CPU altamente integrata, ed un singolo banco di memoria. In questo caso ogni componente viene selezionato e progettato al fine di ridurre il più possibile i costi.

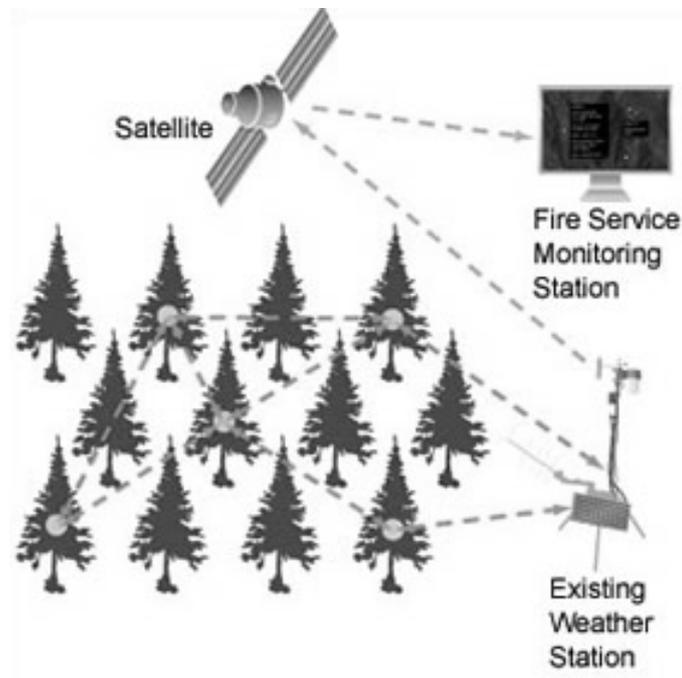


Figura 1.9: Esempio di sistema di misura distribuito per la prevenzione degli incendi

## 1.3 Sistemi di misura distribuiti

### 1.3.1 Il concetto di misurazione distribuita

Il monitoraggio dei fenomeni e delle grandezze di interesse per l'ambiente e per il territorio rappresenta, oggi, una tematica di estremo interesse. In molte applicazioni di importanza scientifico-tecnologico, industriale ed anche sociale, è richiesto il monitoraggio di grandezze dislocate in diverse aree geografiche, per le quali il prelievo manuale dei dati da sensori distribuiti sul territorio, anche se l'area interessata è limitata, risulta antieconomico e, soprattutto, non consente un rilievo tempestivo di eventuali anomalie.

Per effettuare queste misurazioni distribuite si è soliti basarsi su un particolare tipo di sistemi, costituiti da un insieme di dispositivi di misura o di controllo, in grado di svolgere una procedura complessa di misurazione, e posizionati in punti geografici differenti, tra loro connessi attraverso una opportuna tecnologia di comunicazione.

Le possibilità offerte da questo tipo di sistemi di misura sono notevoli; infatti, una volta ultimata la fase di posizionamento dei dispositivi, che può essere effettuata in una qualsiasi zona di interesse, ciascuno degli utenti sarà in grado di accedere ai dispositivi da diverse postazioni, anche se lontane geograficamente. Questo, se da un lato permette di monitorare uno o più processi da vicino pur mantenendosi a distanza, dall'altro fornisce all'utente un accesso diretto ad una strumentazione estremamente costosa, senza però che egli ne venga direttamente a contatto [6].

### 1.3.2 Architettura di comunicazione

La tecnologia hardware di comunicazione può essere di tipo tradizionale, wireless (con l'utilizzo di protocolli quali WiFi o ZigBee) come avviene oramai nella larga maggioranza dei casi, o, in particolari situazioni, adottare soluzioni ibride.

Negli ultimi anni, infatti, l'impiego delle tecnologie wireless ha lentamente soppiantato quelle tradizionali cosiddette *wired*, cioè cablate. I motivi che fanno preferire le tecnologie senza fili, pur se ancora non del tutto standardizzate e caratterizzate da data rate in generale più bassi, sono principalmente l'assenza di cablaggi per l'interconnessione dei dispositivi, una minore estensione geografica della rete, nonché la possibilità di realizzare misurazioni anche in movimento. Come detto, in alcune circostanze si possono avere delle soluzioni ibride che combinano insieme più tecnologie di comunicazione, sia wireless che wired. In questi casi, risultano utili i cosiddetti *bridge* o convertitori di protocollo, che consentono di trasferire i dati tra diversi protocolli mantenendo inalterato il livello di applicazione.

A partire dalla scelta dello specifico protocollo di comunicazione da adottare si determinano l'estensione geografica e la topologia della rete, il numero dei dispositivi impiegati, e le performance raggiungibili in termini di data rate. Se, infatti, l'estensione geografica può andare da qualche metro fino a migliaia di chilometri, per la topologia di rete, nell'ambito delle misure, solitamente

si fa riferimento a due tipi di configurazione: 'a stella' o 'a maglia'. Il data rate è invece un indice del numero di informazioni che possono essere trasmesse in un secondo; esso può variare da poche centinaia di kbps a qualche centinaio di Mbps, a seconda dei casi.

### 1.3.3 Campi di applicazione e sviluppo

I sistemi di misura distribuiti possono venire impiegati in un vastissimo numero di applicazioni, tra i più importanti ricordiamo:

- *monitoraggio dell'inquinamento ambientale*: si utilizza una stazione di rilevamento in grado di eseguire misurazioni relative alla concentrazione nell'aria di inquinanti specifici;
- *monitoraggio di grandezze ambientali*: si realizzano varie misure relative alle condizioni meteorologiche di un dato sito (temperatura, velocità del vento, umidità, pressione);
- *monitoraggio e controllo di una centrale di distribuzione dell'energia elettrica*: è possibile monitorare il funzionamento di tutte le apparecchiature coinvolte;
- *monitoraggio e controllo di una linea di produzione*: tramite l'impiego di dispositivi situati in vari punti della linea si può gestire dell'intero processo da remoto.

Attualmente esistono numerosi sistemi di questo tipo già proposti ed operanti, la cui analisi, però, pone in evidenza alcune problematiche ancora aperte e di rilevante interesse.

Tra queste, le più interessanti da esplorare appaiono le seguenti:

- affidabilità e disponibilità in tempo reale dell'informazione di misura in relazione al monitoraggio di grandezze fortemente suscettibili al rumore ambientale quali livello e portata di corsi d'acqua;
- affidabilità e disponibilità in tempo reale dell'informazione di misura in relazione al monitoraggio di grandezze caratterizzate da elevata dinamica quali la qualità dell'energia elettrica;
- verifica delle prestazioni dei sistemi di comunicazione wireless idonei alla trasmissione dati in applicazioni di monitoraggio di ambiente e territorio;
- caratterizzazione del sistema di monitoraggio nel suo complesso, intesa come caratterizzazione metrologica dei dispositivi di misura e caratterizzazione del sistema di comunicazione ai vari livelli della pila protocollare.

## Capitolo 2

# Idee, motivazioni e prospettive di sviluppo alla base del progetto FLOWEATHER

### 2.1 Idee

#### 2.1.1 Analisi dei dati meteorologici e relativi all'inquinamento

L'idea di base su cui si fonda il progetto FLOWEATHER consiste nella realizzazione di un dispositivo in grado di offrire un monitoraggio costante dell'ambiente all'interno del quale si trova un determinato utente.

Nello specifico, si è preferito focalizzare l'analisi su due grandi famiglie di misure, quelle meteorologiche e quelle riguardanti il grado di inquinamento ambientale, con particolare interesse nei confronti dello studio della qualità dell'aria, senza tuttavia tralasciare altri tipi di inquinamento come quello acustico o elettromagnetico.

La quantità, il numero e la tipologia dei dati estratti possono essere estremamente variabili e dipendono dalle particolari esigenze e dalle diverse situazioni.

Queste misurazioni, effettuate nel tempo, rivestono di per sé una notevole importanza poiché descrivono il comportamento e le variazioni dell'ambiente preso in esame, fornendo al contempo dei validi strumenti per compiere l'analisi di una specifica area localizzata geograficamente.

#### 2.1.2 Integrazione in rete di diversi dispositivi e mappa climatica

Per quanto le misure svolte da una singola stazione appaiano significative, è tuttavia nell'integrazione e strutturazione globale dei dati che sono racchiuse le potenzialità maggiori di un'applicazione di tal genere. Prevedendo, infatti, la realizzazione di numerosi dispositivi, dotati di una comune architettura di rete, si andrebbe a costituire un'infrastruttura di misurazione comune e altamente integrata. In questo modo, sarebbe possibile ottenere una rappresentazione tridimensionale dell'ambiente in cui ciascun dato misurato è caratterizzato, non solo dal suo valore intrinseco ma anche da una dimensione spaziale e una temporale.

Il risultato finale consisterebbe, quindi, nel creare una rete mobile distribuita basata sullo stesso principio delle reti *peer-to-peer*, che consenta agli utenti di condividere dati di qualsiasi genere, e in cui l'elemento base, o nodo della rete, è rappresentato da una piattaforma di sensori per misurazioni ambientali.

La topologia di questa rete si differenzia però da quella utilizzata generalmente nelle reti *peer-to-peer* o di sensori, in cui i nodi sono in grado di comunicare e scambiarsi informazioni reciprocamente. In questo caso, infatti, tutte le misurazioni effettuate da ciascun nodo vengono inviate ad un server centrale, il quale ha lo scopo di memorizzarle, strutturarle e presentarle, rendendole accessibili agli utenti. Si ha perciò una classica topologia 'a stella', in cui il server funge da nodo centrale e svolge la maggior parte del processing dei dati.

I dispositivi collocati nello spazio si comportano come vere e proprie stazioni di misura, operando periodicamente dei rilevamenti tramite i sensori di cui dispongono e trasmettendo i risultati al centro di elaborazione e memoria.

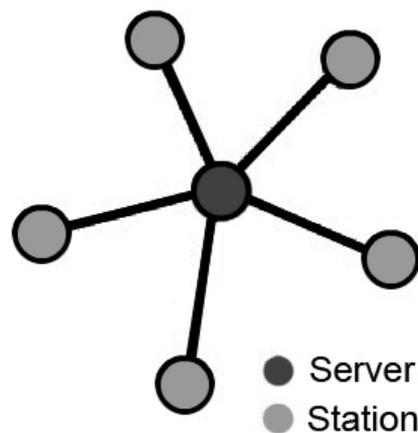


Figura 2.1: Topologia a stella della rete di sensori

### 2.1.3 Presentazione tramite web e personalizzazione del servizio

Una volta che i dati sono stati prelevati, la presentazione finale degli stessi viene effettuata mediante un sito web, che provvede a tracciare una vera e propria mappa delle misurazioni disponibili, ed è affiancata da una serie di grafici ottenuti a partire da opportune segmentazioni operate sui dati. Oltre a questa visualizzazione globale, ciascun utente ha inoltre la possibilità di accedere ad una pagina personale che gli permette di analizzare in maniera specifica i dati raccolti dalla propria stazione.

Questa ambivalenza nel trattamento e nell'utilizzo dei dati, da un lato, permette di sfruttare la ricchezza propria di una database di grandi dimensioni, così da mettere in relazione tutti i tipi di misurazione in base alla loro specifica localizzazione nello spazio, dall'altro, dà la possibilità di fornire al singolo utente un servizio altamente personalizzato e costantemente aggiornato.

#### **2.1.4 Flessibilità da due punti di vista: modularità e mobilità**

Un altro degli aspetti che maggiormente ha contraddistinto il progetto è stato il costante orientamento verso una completa flessibilità, la quale a sua volta si riflette nelle caratteristiche di modularità e mobilità che caratterizzano il sistema. Quest'ultimo, infatti, è pensato come costituito da un unico modulo base, al quale possono essere aggiunti, a seconda delle esigenze, numerosi moduli di estensione, che hanno lo scopo di espandere l'unità centrale, dotandola di funzionalità aggiuntive. Così facendo non si pongono limiti agli sviluppi futuri della piattaforma, che può essere modificata in modo da adattarsi al meglio a seconda delle specifiche finalità che si vogliono perseguire.

La necessità di rendere la stazione mobile è un altro degli obiettivi chiave che si è riusciti a conseguire mediante l'utilizzo di un'interfaccia di comunicazione wireless e di una batteria ricaricabile come forma di alimentazione. La rimozione di ogni vincolo statico, unita alla capacità di autolocalizzazione, contribuisce, infatti, a rendere l'architettura di rete profondamente dinamica e in grado di riorganizzarsi in tempo reale, per far fronte ai continui spostamenti dei nodi.

#### **2.1.5 Consumi energetici ridotti**

Un'ulteriore esigenza è quella di mantenere sotto controllo i consumi energetici, così da permettere una maggiore durata della batteria. Per raggiungere questo obiettivo si è scelto di tenere il dispositivo spento per la maggior parte del tempo pilotandone periodicamente l'accensione tramite un clock, in modo da effettuare le operazioni di misurazione e di successiva trasmissione dei dati rilevati, per poi spegnersi nuovamente. Così facendo, l'intervallo di tempo in cui la stazione è attiva risulta molto breve (dell'ordine di un centinaio di secondi), specie se paragonato al restante periodo di inattività, con la conseguenza di una sostanziale riduzione dei consumi.

#### **2.1.6 Design industriale: la scelta del vaso da fiori**

Notevole importanza, infine, è stata attribuita al profilo estetico del dispositivo in quanto è evidente come un design accattivante faciliti la vendita di un prodotto che, nato per scopi tipicamente scientifici, è in grado di abbinare all'utilità la tipica funzione di un oggetto d'arredamento.

Quest'ultima caratteristica, unita all'impronta fortemente ambientalista che contribuisce a conferire una connotazione di sostenibilità al progetto, ha portato ad una singolare quanto originale scelta per ciò che concerne l'aspetto dell'oggetto in questione. FLOWEATHER, infatti, è stato pensato come un vaso da fiori per esterni, dal design semplice ma innovativo, collocabile su un comune terrazzo ma al contempo in grado di operare le misurazioni per cui è stato progettato.

## 2.2 Motivazioni

### 2.2.1 Un metodo innovativo per misurazioni vicine all'utente

Il massiccio sviluppo sperimentato negli ultimi anni dalle reti di sensori, e in generale da tutte le reti mobili distribuite, ha condotto alla realizzazione di sistemi sempre più funzionali, specifici e maggiormente integrati con l'ambiente all'interno del quale sono inseriti. Tali caratteristiche, grazie alla possibilità di un controllo costante nel tempo, hanno permesso sia di prevedere eventuali fenomeni critici (come ad esempio il rilevamento di incendi o di attività sismiche in una data area a rischio), sia di studiare l'evoluzione temporale di alcune grandezze (come nel caso di sistemi per il controllo delle condizioni climatiche in un determinato habitat naturale o del traffico urbano in una metropoli), in modo da ottenere una conoscenza più approfondita sul comportamento dell'ambiente che ci circonda e con il quale entriamo costantemente in relazione.

Tale innovazione tecnologica ha, quindi, consentito di rivoluzionare il metodo di raccolta, monitoraggio e presentazione dei dati, portando ad una conseguente segmentazione estremamente fine degli stessi, i quali, acquisendo una dimensione personale, ne permettono un utilizzo innovativo e completamente orientato all'utente. Questo concetto si avvicina in parte a quello introdotto dall'avvento delle cosiddette *Personal Area Networks* o *PANs*, le quali, pur con funzionalità profondamente diverse (permettono tipicamente la comunicazione e la sincronizzazione tra molteplici dispositivi vicini a un singolo utente), si sono occupate per prime di definire un'area specifica, di dimensioni ridotte e che pone al centro il singolo utente.

Il progetto FLOWEATHER, sebbene in maniera del tutto differente, si propone di monitorare una porzione d'ambiente che potrebbe essere definita 'in area personale', in quanto strettamente connessa con coloro che operano al suo interno. Ciò è stato pensato allo scopo di fornire una connotazione locale, e quindi specialistica, rispetto ad alcuni tipi di misurazioni, come quelle di natura meteorologica o relative alla qualità dell'aria, caratterizzate generalmente in maniera globale. In tal modo si è provveduto a rendere queste ultime vicine all'utente differenziandole dalle altre disponibili che, essendo lontane, sono al contrario caratterizzate da un minore grado di utilità.

### 2.2.2 Dati meteorologici orari e mappa climatica dei centri urbani

La necessità di disporre di dati meteorologici orari nei centri urbani ha rappresentato, inoltre, una forte spinta allo sviluppo di una stazione portatile in grado di restituire un insieme di dati climatici molto più preciso ed attinente alla località in cui viene prelevato rispetto a quello fornito dalle attuali stazioni meteorologiche, generalmente situate all'interno degli aeroporti o lontano dai centri urbani. L'importanza di avere a disposizione misurazioni climatiche strettamente legate all'ambiente nel quale sono rilevate, e densamente distribuite all'interno di una specifica area d'interesse, si riflette nella capacità di realizzare una mappatura del clima estremamente precisa, la quale, a partire dai dati raccolti dalle stazioni poste in diversi punti dello spazio,

permetta di apprezzare al meglio le variazioni, anche sostanziali, che avvengono tra zone vicine nella stessa porzione di spazio osservata. Se si prende in esame, ad esempio, una realtà urbana variegata come quella della città di Palermo, appare evidente come, spostandosi da una zona molto raccolta e densamente edificata tipica del centro storico verso una più aperta come il porto cittadino, avvengano dei cambiamenti climatici di una certa rilevanza, sebbene le due aree prese in esame risultino poco distanti tra loro.

Tutto ciò contribuirebbe in maniera determinante ad accrescere le potenzialità relative all'ambito degli studi climatologici e di quelli urbanistici. Sarebbe in tal modo soddisfatta l'esigenza di disporre di dati inerenti al clima che siano precisi e attendibili, il che costituisce un vincolo imprescindibile dell'analisi territoriale e della progettazione architettonica, la quale, sia essa orientata all'edilizia che alle infrastrutture, deve attingere informazioni da una specifica fonte di dati climatici, specie se si propone come obiettivo quello di un'ottimizzazione energetica esecutiva del progetto.

Una progettazione energeticamente consapevole, infatti, deve tenere conto dei parametri climatici favorendo, ove possibile, lo sfruttamento di quelli che possono fornire un contributo positivo al bilancio energetico dell'edificio e migliorare nel contempo le condizioni di comfort. Si definisce comfort ambientale quella particolare condizione di benessere determinata, in base alle percezioni sensoriali di un individuo inserito in un ambiente, da temperatura, umidità dell'aria, livello di rumorosità e luminosità, tutte condizioni in gran parte pianificabili e di responsabilità del progettista. Da tale definizione si ha una distinzione tra benessere termo-igrometrico, benessere acustico e benessere luminoso.

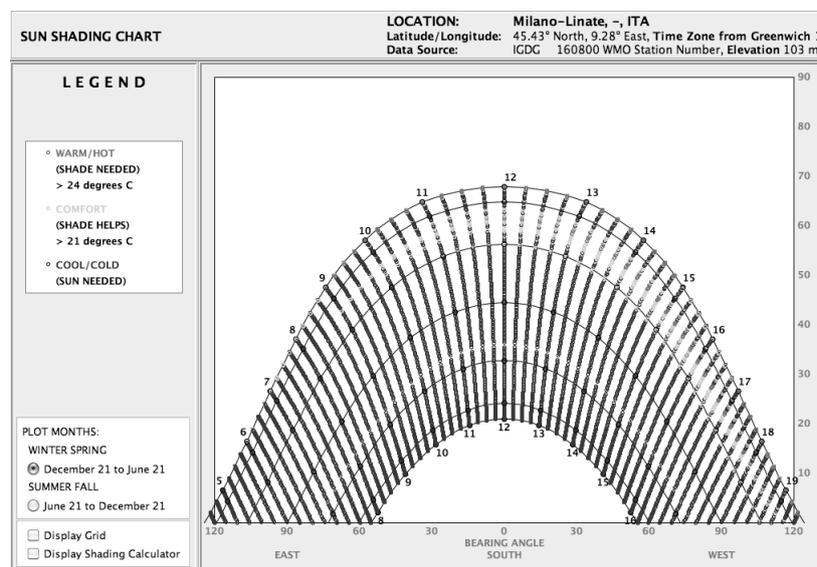


Figura 2.2: Schermata di Climate Consultant 4: grafico di ombreggiatura solare ricavato a partire ai dati climatici orari raccolti nella zona di Milano Linate nel 2009

## 2.2.3 Uno strumento per la gestione energetica degli edifici

Un'ulteriore utilità è rappresentata dalla capacità di offrire un valido supporto al successivo processo di gestione energetica degli edifici, che si fonda su una politica di riduzione dei consumi mirata ad essere quanto più efficiente possibile. Ciò è ottenibile attraverso una valutazione dei sistemi termoigrometrici, acustici, luminosi e di qualità dell'aria, al fine di realizzare una gestione del controllo ambientale, considerando l'assunto che il risparmio energetico debba essere sempre eseguito in presenza, e non a discapito, delle condizioni di comfort e benessere negli ambienti serviti da sistemi impiantistici.

Sotto questo punto di vista FLOWEATHER appare lo strumento ideale per soddisfare queste esigenze, in quanto è in grado di sostituirsi completamente ai molteplici sistemi responsabili di tutti i tipi di misurazione così da poter essere impiegato come unico dispositivo demandato all'estrazione dei dati orientati al controllo dell'efficienza.

Un'altra importante funzionalità è quella che permette l'utilizzo dei dati raccolti all'interno di software specifici di progettazione che effettuano simulazioni energetiche. Ne sono esempi *EnergyPlus* [7], strumento che rientra nel programma del dipartimento energetico degli Stati Uniti per la progettazione sostenibile degli edifici [8], o *Climate Consultant* [9], il cui obiettivo è di mostrare speciali grafici a partire dai dati climatici della località scelta, evidenziando il pattern complessivo e i dettagli dei diversi climi che potrebbero altrimenti andare perduti in normali tabelle numeriche. Ciò è legato alla scelta di rendere disponibili i dati immagazzinati non solo tramite un'interfaccia web ma anche all'interno di un documento specifico, scaricabile gratuitamente, e aggiornato all'ultima misurazione effettuata. Tale file risulta essere in formato EPW, una speciale estensione che racchiude dati climatici ed è utilizzata da numerosi software, come quelli descritti in precedenza [10].

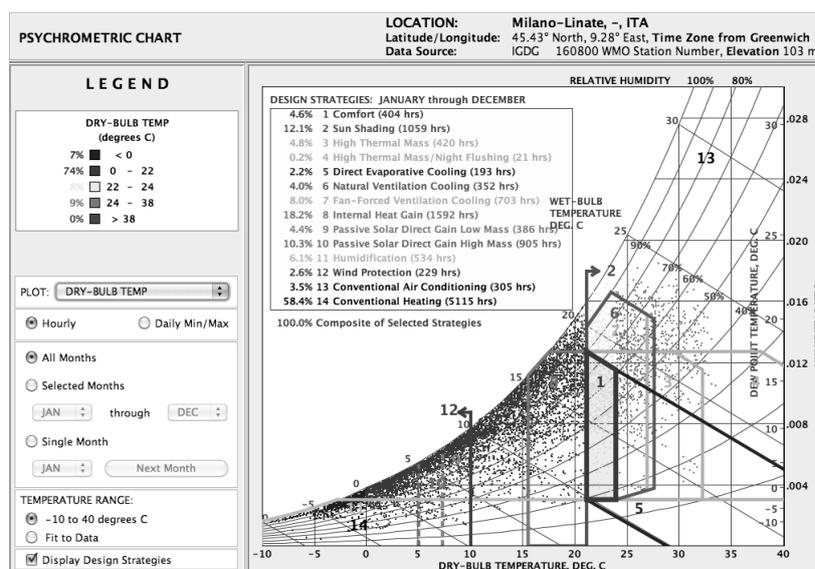


Figura 2.3: Schermata di Climate Consultant 4: Consigli sulle strategie di progetto ricavati in base ai dati climatici orari misurati nella zona di Milano Linate nel 2009

## 2.2.4 Il monitoraggio ambientale dal punto di vista dell'inquinamento

Le stesse potenzialità esaminate in ambito climatico, possono essere ugualmente sfruttate nell'analisi dell'impatto prodotto da vari tipi di inquinamento, come quelli atmosferico, acustico ed elettromagnetico. Anche in questo caso, quindi, si avrebbe la possibilità di creare una precisa mappa della qualità dell'aria o dell'inquinamento acustico presente in un dato centro urbano, così da confrontare i dati con quelli forniti periodicamente dai comuni, verificandone la validità. Risulterebbe allora interessante sapere in quali zone di una determinata città l'aria è più respirabile, o se un appartamento è situato in una zona particolarmente caotica. Gli utenti sarebbero in questo modo capaci di controllare autonomamente il grado di inquinamento dell'ambiente in cui vivono, e, attraverso la condivisione di queste informazioni con altri, di riuscire a prendere coscienza degli eventuali rischi per la salute, ed essere in grado di porvi rimedio, al fine di migliorare la qualità della propria vita.

Occorre inoltre ricordare come il costante monitoraggio dei livelli di inquinamento dell'aria sia ormai divenuto una necessità per gli enti provinciali e regionali all'interno della Comunità Europea, così come la conseguente responsabilità di mantenere tali livelli sotto la soglia cosiddetta 'di informazione'.

## 2.2.5 Originalità e innovatività

In conclusione, è importante sottolineare come la motivazione principale alla base di FLOWEATHER sia complessivamente riconducibile all'originalità del progetto, testimoniata dall'assenza sul mercato odierno di un dispositivo con tali caratteristiche.

Le prerogative di totale flessibilità e mobilità, accompagnate da altre importanti specifiche quali le piccole dimensioni, i consumi, i costi ridotti ed il design efficace, contribuiscono infatti a rendere un tale prototipo unico nel suo genere.

Innovativo risulta inoltre l'approccio adottato per la presentazione dei dati, che è stato studiato per risultare intuitivo ed a misura di utente. Basandosi infatti sul modello dei moderni *social networks*, ci si propone di ampliare la conoscenza che ciascuno possiede riguardo all'ambiente in cui si trova, accompagnando ai diagrammi usati per offrire una rappresentazione grafica dei dati raccolti, un insieme di commenti e suggerimenti con lo scopo di fornire un quadro preciso del luogo abitato e, al contempo, migliorare il proprio confort.

## 2.3 Prospettive di sviluppo

### 2.3.1 Commercializzazione del prodotto: l'ambito consumer

La possibilità di arrivare alla commercializzazione, e in seguito alla distribuzione, di un prodotto con le caratteristiche di FLOWEATHER, appare uno scenario realistico, sia, come già sottolineato in precedenza, a causa dell'assenza sul mercato di un dispositivo che assolva determinate funzioni, sia per il crescente interesse che si sta sviluppando attorno al tema della salvaguardia dell'ambiente e della salute degli individui che lo abitano.

La distribuzione su larga scala porterebbe, infatti, alla creazione di una rete sociale di utenti, costituita principalmente da persone sensibili a tematiche ambientaliste, in grado di entrare in contatto e cooperare condividendo i dati a disposizione e scambiandosi, al contempo, informazioni di vario genere, così da produrre una connessione tra gli utenti proattivi all'interno della rete. Questo approccio, che è alla base dei più noti social network, è infatti applicabile con successo anche al progetto FLOWEATHER, e permetterebbe, agli utenti che hanno acquistato la stazione, di registrarla sul sito *floweather.com* in modo da poter visionare costantemente i dati raccolti, creando allo stesso tempo un profilo personale e una lista di contatti.

Il progetto si fonda inoltre sulla sostenibilità, sia dal punto di vista dell'oggetto in sé, il quale è ideato per ricaricarsi autonomamente, sia in un'ottica di più ampia scala che coinvolge gli utenti che hanno deciso di supportare il progetto stesso. L'aggregazione complessiva dei dati, infatti, sempre nell'ipotesi di una notevole penetrazione, rappresenta probabilmente la risorsa più importante a disposizione poiché può, nella sua totalità, essere messa a disposizione di tutti coloro interessati, siano essi rappresentati da aziende, enti o amministrazioni pubbliche. In tal modo, sarà possibile sottoscrivere un abbonamento che permetta di fruire dell'intero database contenente le misurazioni operate da tutte le stazioni nell'arco di un anno. I proventi ricavati dalla vendita dei dati complessivi verranno poi ridistribuiti equamente tra tutti gli utenti i quali avranno così modo di recuperare in parte l'investimento iniziale effettuato per acquistare la stazione.

### 2.3.2 Un secondo possibile utilizzo: l'analisi scientifica

Le previsioni circa i possibili sviluppi futuri del dispositivo in questione non riguardano tuttavia unicamente la prospettiva del singolo utente che decide di dotarsi di uno strumento di misura per scopi prettamente domestici.

Accanto a questo possibile utilizzo, basato più sul numero delle misurazioni raccolte che sulla qualità delle stesse, ne esiste un altro, diametralmente opposto al primo e prettamente orientato alla raccolta e all'analisi scientifica dei dati. Supponendo, ad esempio, di sovrapporre alla rete costituita dalle stazioni meteorologiche già presenti sul territorio una seconda rete *overlay*, costituita da dispositivi meno precisi ma distribuiti in maniera più densa e numerosa, si andrebbe

a potenziare l'attività di rilevazione e ad incrementarne in maniera sostanziale la diffusione sul territorio.

Un'altra peculiarità è costituita dalla possibilità di impiegare tali dispositivi come nodi di una comune rete wireless di sensori, in maniera perciò assestante rispetto a quanto visto prima, per scopi specifici di tipo proprietario; come, ad esempio, nel caso di un gruppo di ricerca interessato a studiare le variazioni dei venti in un dato tratto costiero o, ancora, ad analizzare le variazioni climatiche che avvengono durante l'anno in una specifica area di macchia mediterranea.

# Capitolo 3

## Progettazione e realizzazione del prototipo

### 3.1 Definizione dei driver di progetto

Le fasi di progettazione e successiva realizzazione del prototipo FLOWEATHER sono state condotte tenendo costantemente in considerazione numerosi aspetti, molti dei quali sono alla base dell'idea che ha ispirato il lavoro, quella cioè di implementare un dispositivo innovativo, utile e accessibile a tutti.

La filosofia adottata può infatti essere ben sintetizzata dalla definizione di '*GLOSS project*' (*Green, Low cost, Open Source, Sustainable*), acronimo che contiene i quattro principali driver di progetto che hanno caratterizzato l'operato, e rispetto ai quali è stata orientata la maggior parte dell'attività progettuale.

#### 3.1.1 La prospettiva *Green*

Quella dell'impegno ambientale, che è forse la caratteristica primaria del progetto, è testimoniata dallo sforzo prodotto per realizzare un sistema per il monitoraggio dell'ambiente capace di rilevare tutte le particolari tipologie di dati che influenzano la qualità della vita degli individui che lo popolano.

Essa è inoltre rafforzata dalla precisa scelta di utilizzare batterie ricaricabili per garantire l'alimentazione e materiali altamente riciclabili come il polietilene per le parti esterne del dispositivo, al fine di ridurre al massimo la quantità di sostanze inquinanti potenzialmente disperdibili nell'ambiente.

Il polietilene (più comunemente noto come polietilene) è il più semplice tra i polimeri sintetici ed il più comune fra le materie plastiche; viene spesso indicato con la sigla 'PE'. Il polietilene è una resina termoplastica, si presenta come un solido trasparente (forma amorfa) o bianco (forma cristallina) con ottime proprietà isolanti e di stabilità chimica, è un materiale molto versatile ed una delle materie plastiche più economiche; gli usi più comuni sono come isolante per cavi elettrici, film per l'agricoltura, borse e buste di plastica, contenitori di vario tipo, tubazioni, strato interno di contenitori asettici per liquidi alimentari e molti altri. Oltre a essere longevo

esso è un materiale che risulta adeguato ad essere riciclato dopo il suo utilizzo: i composti del polietilene sono infatti riciclabili al 100% fisicamente, chimicamente o energeticamente. I prodotti del polietilene hanno una vita utile molto lunga e ciò diminuisce la produzione di rifiuti se usati al posto di prodotti meno durevoli [11].

### 3.1.2 La prospettiva *Low cost*

La capacità di mantenere i costi bassi ha rappresentato un'altra affascinante sfida affrontata durante la fase di pianificazione. L'obiettivo iniziale di contenere i costi totali, mantenendoli al di sotto della soglia del centinaio di euro, non è però apparso realisticamente perseguibile a causa dell'elevato numero di componenti prodotti da aziende diverse che è stato necessario assemblare; tuttavia il risultato raggiunto ha portato ad un costo complessivo che si attesta attorno ai centocinquanta euro.

Questo appare un dato confortante specie se paragonato a quello di altre stazioni meteorologiche pensate per ambiti casalinghi, i cui prezzi di vendita risultano sensibilmente superiori. Occorre inoltre specificare che, trattandosi di un prototipo, i costi sostenuti per realizzare una sola unità sono stati nettamente superiori rispetto a quelli che si sarebbero dovuti affrontare nel caso di produzione industriale in serie.

### 3.1.3 La prospettiva *Open Source*

Un altro concetto forte, che ha rappresentato un'importante spinta allo sviluppo del progetto, è legato alla filosofia open source, che è stata sposata integralmente e in maniera trasversale tanto per il software quanto per l'hardware.

FLOWEATHER è infatti basato su Arduino, una piattaforma di prototipazione elettronica open source, basata su hardware e software flessibili e semplici da utilizzare.

Gli schemi hardware di Arduino sono distribuiti in modo da poter essere utilizzati nei termini legali di una licenza *Creative Commons Attribution Share-Alike 2.5* [12] e sono disponibili sul sito ufficiale *arduino.cc*. Il codice sorgente per l'ambiente di sviluppo integrato e la libreria residente sono disponibili, e concessi in uso, secondo i termini legali contenuti nella licenza *GPLv2*.

Tutto il materiale prodotto per l'occasione (sketch, librerie, file) è reso accessibile, consultabile e scaricabile gratuitamente dal dominio *floweather.com*.

Così facendo, viene data la possibilità, a chiunque fosse interessato, di entrare in possesso degli strumenti necessari per realizzare autonomamente il prototipo o, in alternativa, di apportare delle modifiche allo stesso in modo da renderlo utile per altri scopi.

La condivisione viene perciò vista come un'opportunità per migliorare ulteriormente il prodotto, basandosi sull'esperienza e sulle idee di chiunque abbia voglia di aiutare a far progredire il progetto.

### 3.1.4 La prospettiva *Sustainable*

L'ultimo requisito che si è scelto di prendere in considerazione riguarda la sostenibilità; esso si riflette in due differenti aspetti.

Il primo, affrontato specificatamente dal punto di vista dell'utente, mira al recupero nel tempo dell'investimento effettuato acquistando il dispositivo. Ciò è ottenibile dalla vendita dei dati complessivi, misurati annualmente da tutte le stazioni e raccolti in un unico documento. I proventi così ricavati verrebbero successivamente suddivisi in egual misura tra tutti i possessori dei dispositivi, in modo da contribuire alla sostenibilità economica della piattaforma. Riguardo all'attrattività di un tale insieme aggregato di misurazioni si è già discusso a sufficienza nei paragrafi precedenti.

Il secondo aspetto, al contrario, si concentra maggiormente sulla progettazione hardware, e ha permesso di rendere il prototipo sostenibile energeticamente. Tramite l'impiego di un pack di batterie ricaricabili agli ioni di litio, unito alla realizzazione di un modulo per l'alimentazione basato su un pannello solare, infatti, si è stati in grado di creare un dispositivo autonomo ed a risparmio energetico.

La sostenibilità è infine accentuata dalla volontà di ridurre al massimo l'impatto sull'ambiente. Per tal motivo sono stati impiegati unicamente materiali altamente riciclabili, oltre a componenti elettronici che rispettano la normativa RoHS che vieta l'uso di sostanze nocive quali piombo, mercurio e cadmio.

### 3.1.5 Una doppia direttrice di sviluppo: *floweatherpot* e *floweather.com*

Lo sviluppo del progetto FLOWEATHER ha seguito due piani paralleli, profondamente diversi ma complementari.

Il primo ha riguardato l'implementazione del dispositivo vero e proprio, sotto forma di stazione mobile dotata di alta flessibilità ed a basso consumo. A partire da queste caratteristiche è stato necessario scegliere con attenzione e assemblare i numerosi componenti per svolgere le funzioni di sensing, comunicazione e gestione consumi. Notevole attenzione è stata inoltre attribuita al design industriale del prototipo, concepito come un piccolo vaso da fiori per esterni, che, proprio grazie a questa sua funzione alternativa, prende il nome di *floweatherpot*.

Il secondo piano è invece consistito nell'ideazione di una struttura informatica a supporto del cosiddetto lato server, che ha lo scopo fondamentale di memorizzare al suo interno i dati inviati dalle diverse stazioni e presentarli ai rispettivi utenti. Ciò è fatto al fine di realizzare un servizio semplice, altamente personalizzato, capace di lavorare in tempo reale e del tutto automatizzato.

Tale struttura si appoggia sul sito web *floweather.com*, che rappresenta l'interfaccia grafica attraverso cui è possibile visionare i dati misurati dalla propria stazione, strutturati in opportuni grafici, e condividerli con altri utenti. Anche in questo caso, si è data molta importanza al web design, cercando di utilizzare un layout essenziale e gradevole allo stesso tempo.

## 3.2 Il dispositivo: *floweatherpot*

La maggior parte del lavoro svolto per la realizzazione del prototipo è stata rivolta all'implementazione della stazione vera e propria, dal momento che è stato necessario soddisfare diversi requisiti, sia hardware che software.

A partire dagli obiettivi che si intendevano raggiungere e in base ai driver individuati in sede di progetto, è apparso indispensabile scegliere quei particolari componenti che, una volta assemblati, garantissero determinate caratteristiche e permettessero lo svolgimento di determinate funzionalità.

### 3.2.1 Caratteristiche

Le proprietà distintive che caratterizzano in maniera sostanziale il dispositivo sono quattro:

1. Mobilità;
2. Modularità;
3. Hardware libero;
4. Flessibilità.

Ciascuna di esse risulta infatti fondamentale a conferire al prototipo un'identità altamente versatile, ma al contempo specifica.

#### 3.2.1.1 Mobilità

La mobilità rappresenta probabilmente la prerogativa più importante tra quelle individuate. Essa si riflette nella necessità, da un lato di dotare la stazione di un'interfaccia di comunicazione di tipo wireless, dall'altro di usare un pack di batterie per alimentare la stessa, dal momento che non è prevista la presenza di alcun cablaggio che limiti il posizionamento e la futura mobilità del dispositivo.

Rimuovendo in questo modo ogni vincolo legato alla staticità della piattaforma, si ottiene una libertà assoluta rispetto alla dimensione spaziale, che la rende quindi collocabile in qualsiasi punto e successivamente riposizionabile.

#### 3.2.1.2 Modularità

Altrettanto caratterizzante appare la propensione nei confronti di un'ottica improntata alla totale modularità. Lo schema di implementazione della stazione è stato infatti seguito prevedendo l'eventualità di realizzare un'espansione della piattaforma, così da consentire lo sviluppo di nuove

funzionalità all'interno della struttura preesistente. L'integrazione avviene attraverso l'aggiunzione di moduli supplementari che, connettendosi a quello base, permettono di ampliare la tipologia di misure effettuate o di abilitare nuovi meccanismi di comunicazione basati su protocolli alternativi. In questo modo l'unità centrale potrà interfacciarsi direttamente con le estensioni al fine di ricevere o trasmettere informazioni.

Questa peculiarità potrebbe anche venire adoperata in ambito commerciale per far leva sull'attrattiva del prodotto; essa invoglierebbe infatti gli utenti ad comporre la propria stazione ideale aggiungendo di volta in volta i moduli che ciascuno preferisce.

### **3.2.1.3 Hardware libero**

Con il termine open hardware si è soliti indicare quei particolari tipi di hardware elettronici progettati seguendo la stessa politica del software libero open source (*Free Open Source Software* o *FOSS*). L'hardware libero è infatti parte della cultura dell'open source, che espande quest'ideologia al di fuori dell'ambito del software. Il termine viene principalmente usato per esprimere la libera divulgazione di informazioni riguardanti il progetto stesso dell'hardware, comprendente gli schemi, la lista dei materiali, il layout dei dati del circuito stampato, solitamente insieme al FOSS per far girare l'hardware.

La scelta di impiegare la piattaforma Arduino come elemento fondante del dispositivo rientra appunto in questa filosofia di condivisione e divulgazione dell'operato scientifico, volta a permettere ad altri sviluppatori di realizzare il medesimo sistema o a prenderlo come spunto per la progettazione di un altro prototipo originale basato sulle stesse caratteristiche.

### **3.2.1.4 Flessibilità**

Il concetto di flessibilità può essere visto come una summa delle tre proprietà precedentemente esaminate. Essa è infatti intesa come versatilità, sia riguardo alla capacità di posizionare il dispositivo in assoluta libertà o di ampliarne le funzionalità mediante l'impiego di moduli aggiuntivi, sia nella prospettiva di uno sviluppo futuro fondato sulla partecipazione da parte di gruppi di utenti in grado di fornire nuovi contributi al progetto.

*Flowerpot* si configura perciò come una stazione estremamente duttile, trasformabile in base alle esigenze e impiegabile anche in applicazioni mobili che richiedono misurazioni di tipo dinamico all'interno dell'ambiente di riferimento.

L'opportunità di effettuare un'estensione e di abilitare funzionalità extra consentirebbe di ideare qualsiasi tipo di modulo da annesso al sistema, così da renderlo capace di raggiungere target per i quali non è stato inizialmente concepito.

## 3.2.2 Componenti

Il processo di individuazione e selezione dei componenti da assemblare per comporre il prototipo è stato influenzato dalla ricerca di particolari elementi qualitativamente idonei ma allo stesso tempo in grado di mantenere i costi al di sotto della soglia prestabilita.

Per raggiungere questa finalità è stato necessario scegliere componenti elettronici di buona fattura che, ripetutamente testati, garantissero un comportamento altamente affidabile.

Il modulo centrale è costituito da una scheda *Arduino Duemilanove* a cui sono connessi un modulo GSM/GPRS, tre diversi tipi di sensori ed un real-time clock con lo scopo di mantenere sincronizzato l'apparato anche quando esso è spento.

La stazione è inoltre equipaggiata con un pack di batterie di tipo LiIon e uno schermo LCD utilizzato per visualizzare le ultime misurazioni.

### 3.2.2.1 Scheda Arduino Duemilanove

Una scheda Arduino consiste in un microcontrollore a 8-bit *AVR* sviluppato da *ATmel*, con l'aggiunta di componenti complementari che ne facilitino l'incorporazione in altri circuiti. Le Arduino Duemilanove usano chip della serie *megaAVR*, modello *Atmega328* [13].

Ciascuna board possiede i seguenti elementi:

- quattordici piedini digitali input/output digitali (di cui 6 utilizzati come output PWM);
- sei input analogici;
- un oscillatore a cristallo da 16MHz;
- un connettore USB;
- un jack di alimentazione;
- un ICSP header (In-Circuit Serial Programming);
- un bottone di reset.

La scheda Arduino Duemilanove può essere alimentata tramite connessione USB oppure tramite alimentatore di corrente. La sorgente di alimentazione è selezionata automaticamente. L'alimentazione esterna (non USB) può arrivare sia da un alimentatore AC-DC oppure da una pila.

Il microprocessore *Atmega328* possiede 32kB di memoria flash per caricare il codice (di cui 2kB sono utilizzati per il *bootloader*). Ha 2kB di SRAM e 1024Bytes di *EEPROM*.

Il software utilizzato è un'applicazione multiplatforma scritta in Java, ed è derivata dall'IDE creato per il linguaggio di programmazione *Processing* e per il progetto *Wiring*.

L'ambiente di sviluppo integrato di Arduino è fornito di una libreria software C/C++ chiamata 'Wiring', che rende molto più semplice implementare via software le comuni operazioni input/output. I programmi di Arduino sono scritti in C/C++, ma, per poter creare un file eseguibile, all'utilizzatore non è chiesto altro se non definire due funzioni:

- `setup()`: una funzione invocata una sola volta all'inizio di un programma che può essere utilizzata per i settaggi iniziali;
- `loop()`: una funzione chiamata ripetutamente fino a che la scheda non viene spenta.

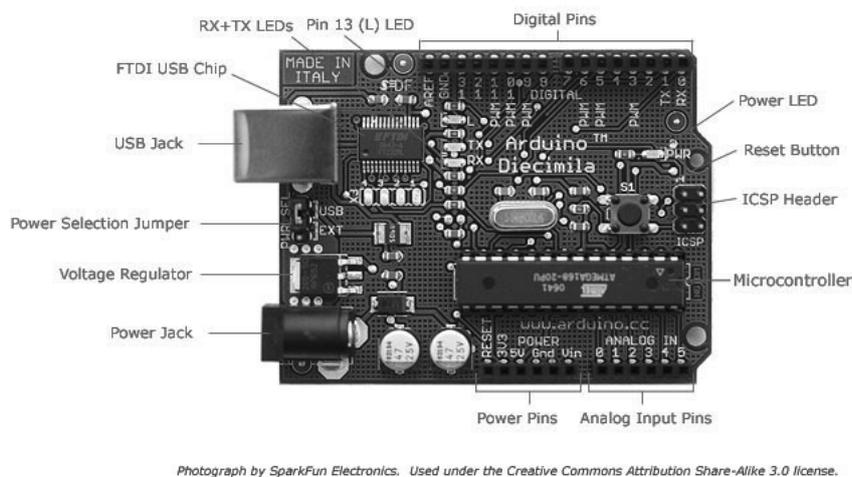


Figura 3.1: Scheda Arduino con evidenziati i diversi componenti

### 3.2.2.2 Sensori

Le misurazioni che si sono scelte di effettuare riguardano la temperatura, la pressione atmosferica, l'umidità e la concentrazione di monossido di carbonio (CO) presenti nell'aria.

L'estrazione di tali dati è realizzata grazie all'impiego di tre diversi sensori:

- SHT15: prodotto da Sensirion [14], appartiene alla famiglia di sensori di temperatura e umidità relativa SHT1x. E' basato sulla tecnologia CMOS che garantisce un'eccellente affidabilità, oltre ad una stabilità nel lungo periodo. Entrambe le misurazioni una volta effettuate analogicamente vengono convertite e rese disponibili attraverso un output digitale. Il sensore fornisce inoltre una calibrazione automatica, grazie ad una serie di coefficienti preprogrammati inseriti nel chip che permettono una rilevamento preciso dei segnali, oltre a misurare il grado di umidità in funzione della temperatura calcolata. Esso è connesso alla scheda madre attraverso un'interfaccia seriale a due uscite, e grazie ad un regolatore interno del voltaggio risulta rapido e dotato di grande reattività. Le dimensioni ridotte insieme agli scarsi consumi lo rendono infine il candidato ideale per un tale tipo di applicazione.

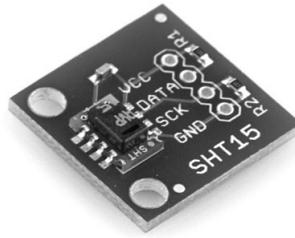


Figura 3.2: Sensore SHT15

- MPX4115A: scelto per misurazioni di pressione, questo sensore integrato su un solo circuito è in grado di compensare la pressione atmosferica assoluta in base alla temperatura offrendo una precisa calibrazione. Fornisce la lettura del valore stimato attraverso un output analogico e risulta sufficientemente preciso (errore massimo stimato nell'ordine dello 1.5%). La sua semplicità d'uso unita al prezzo vantaggioso lo rendono uno dei trasduttori più utilizzati in applicazioni per il rilevamento climatico. E' distribuito da Motorola [15].



Figura 3.3: Sensore SHT15

- TGS2442: sensore realizzato da Figaro Sensors [16] per il rilevamento del monossido di carbonio, è caratterizzato da alta sensibilità, bassi consumi, piccole dimensioni e bassa dipendenza dall'umidità. Lavora per mezzo di un circuito elettrico che fornisce una pulsazione costante per ogni ciclo di voltaggio della durata di un secondo, convertendo il cambio di conduttività nel segnale di output, cui corrisponde la concentrazione del gas misurata nell'ambiente. A partire dal valore di resistenza misurato è perciò possibile, tramite lo studio della caratteristica di sensibilità, risalire al valore comunemente impiegato, valutato in ppm (parti per milione), e trasmesso mediante un output analogico.



Figura 3.4: Sensore TGS2442

### 3.2.2.3 Modulo GSM/GPRS

Il GM862-Quad è un modulo per comunicazioni mobili sviluppato da Telit [17]. E' basato sullo standard GSM, di cui supporta le quattro bande di frequenza utilizzabili (850, 900, 1800 e 1900 MHz), inoltre abilita l'impiego servizi di trasmissione di dati a pacchetto attraverso la tecnologia GPRS integrando la gestione dello stack protocollare TCP/IP. Tra le principali caratteristiche, la possibilità di inviare, ricevere o processare SMS, di attivare una connessione con un server tramite l'apertura di un socket, di comunicare con la cella a cui è connesso per ottenere informazioni relative alla propria localizzazione.

Necessita di essere equipaggiato con un SIM card da inserire nell'alloggiamento apposito, per abilitare le funzionalità di invio di SMS e comunicazione con il server, che rappresentano i due metodi previsti per trasmettere i dati raccolti dalla stazione.

Per ottenere una migliore connettività occorre infine collegare il modulo ad una antenna GSM QUAD band; nello specifico ne è stata scelta una di piccole dimensioni (34x7mm) prodotta da Taoglas.



Figura 3.5: Modulo GM862

### 3.2.2.4 Real time clock

L'idea di un dispositivo che permetta di mantenere quanto più bassi possibili i consumi è incentrata sulla capacità di limitare il comportamento dello stesso unicamente alle fasi di sensing e comunicazione dei dati, mantenendo la stazione spenta per il restante tempo.

Per riuscire in tale intento, si è dovuto inserire un componente che garantisca questa funzione di sincronizzazione.

Il Real Time Clock BV4236, realizzato da ByVac [18], consente, tramite la gestione di un segnale di interrupt, di risvegliare il sistema dalla modalità di 'sleep' in cui esso normalmente si trova in modo da permettergli di svolgere le funzionalità standard prima di spegnersi nuovamente.

L'oscillatore al quarzo contenuto nel clock, genera infatti un segnale periodico che permette di tenere traccia dell'orario anche quando tutti gli altri componenti sono inattivi, e ridestare così la stazione ad intervalli periodici ogni ora. Ciò si riesce ad ottenere mediante la presenza di una batteria di backup che mantiene attivo il quarzo anche in assenza di alimentazione.

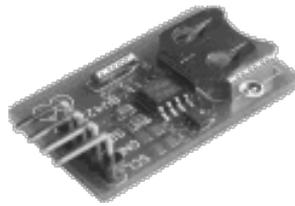


Figura 3.6: RTC BV4236

### 3.2.2.5 Display LCD

Un'altra modalità per la visualizzazione dei dati è quella che dà l'opportunità a qualsiasi utente di leggere l'ultima misurazione effettuata direttamente dal dispositivo. Esso è infatti dotato di un piccolo schermo a cristalli liquidi (LCD) attivabile a comando mediante un pulsante, che indica data, ora e valori presenti in memoria, i quali corrispondono a quelli dell'ultima misura in ordine di tempo.

Per questioni legate al risparmio energetico, il display è stato programmato in modo da restare acceso per cinque secondi, esso inoltre non risulta attivabile se la stazione è in fase di misurazione.

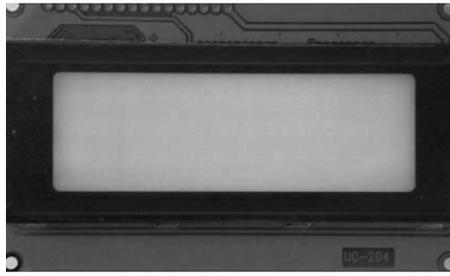


Figura 3.7: Display LCD

### 3.2.2.6 Batterie e circuito di ricarica

Il dispositivo viene alimentato grazie ad un pack costituito da una coppia di batterie ricaricabili agli ioni di litio (Lion), da 3,7V e 2000mAh ciascuna, che, collegate in serie, forniscono una tensione complessiva di 7,4V.

Il circuito di ricarica e monitoraggio esegue un controllo costante sulla carica residua a disposizione avvisando l'utente, tramite l'accensione di un led, quando essa scende al di sotto della soglia critica, e necessita perciò di essere reintegrata.

Il ripristino della carica può avvenire attraverso l'impiego di un normale trasformatore, o con un modulo supplementare, costituito da un piccolo pannello solare studiato appositamente per rendere il sistema indipendente e completamente autosufficiente. Esso consentirebbe inoltre di aggiungere un'altra misurazione a quelle già presenti, quella relativa alla radiazione solare.



Figura 3.8: Pack di batterie agli ioni di litio

Nome	Categoria	Funzione	Produttore	Pin	Protocollo
Arduino 2009	Scheda madre	Microcontrollore	Arduino	14D + 6A	-
SHT15	Sensing	Temperatura e Umidità	Sensirion	2D	Digital 2 wire
MPX4115	Sensing	Pressione atmosferica	Motorola	1A	Analog 1 wire
TGS2442	Sensing	Monossido di carbonio	Figaro	2A	Analog 1 wire
BV4236	Sincronizzazione	Real Time Clock	ByVac	1A + 2D	I2C
GM862 QUAD	Comunicazione	Modulo GSM/GPRS	Telit	2A + 1D	Serial
GM862 EB 50pin	Comunicazione	Board per GM862	Sparkfun	-	-
PCB Mini 34x7mm	Comunicazione	Antenna GSM QUAD	Taoglass	-	-
SIM Card	Comunicazione	Mobile Equipment	Wind	-	-
HD44780	Visualizzazione	Display LCD 4x20	Orient Display	7D	Hitachi
PCF8574A	Visualizzazione	Espansore 8 porte	Texas Instruments	2A	I2C
Lion Battery Pack	Ricarica	Coppia batterie a ioni di litio	Enix Energies	-	-
1.3W 10V Solar Cell	Ricarica	Pannello solare	Voltaic	-	-

Tabella 3.1: Panoramica dei componenti impiegati

### 3.2.3 Funzionalità

La piattaforma *floweatherpot* è stata progettata per effettuare misurazioni con frequenza oraria ed inviarle successivamente ad un server attraverso una connessione GPRS. Una volta ultimate queste attività il sistema si arresta per riattivarsi automaticamente trascorsi sessanta minuti.

I compiti svolti possono perciò essere raggruppati in due grandi categorie, relative rispettivamente alla raccolta e alla trasmissione di informazioni, entrambe studiate in modo da consumare la minore quantità di energia possibile.

#### 3.2.3.1 Sensing

I tre sensori utilizzati sono disposti sulla stessa linea di alimentazione, la quale normalmente appare aperta grazie all'impiego di un transistor. Ogni qualvolta scatta l'allarme orario, il clock provvede a risvegliare il dispositivo, la linea viene allora cortocircuitata ed inizia la fase di rilevazione.

I sensori SHT15 e MPX4115, dal momento che non necessitano di un transitorio prima di andare a regime, eseguono unicamente un ciclo di undici misurazioni, una ogni mezzo secondo, per poi selezionare il valore mediano all'interno di ciascun vettore stimato. In questo modo si riesce a proteggere il sistema nel caso, seppur raro, di una singola misura errata. Temperatura, umidità e pressione atmosferica risultano quindi facilmente calcolabili; più complesso appare invece il meccanismo legato alla quarta tipologia di misurazione, ovvero quella riguardante la concentrazione di monossido di carbonio nell'ambiente.

Il sensore TGS2442, infatti, a causa della maggiore complessità del dato da rilevare, è caratterizzato da un comportamento più articolato. Esso è costituito da una coppia di resistenze connesse in serie: la prima rappresenta il carico e ha lo scopo di alimentare la seconda, che a sua volta funge da elemento di sensing. Alimentate opportunamente con due diversi cicli di voltaggio, i due resistori forniscono una lettura che, corretta mediante un parametro che tiene conto della temperatura, permette di ottenere la conversione al valore cercato.

Per questo motivo il sensore ha bisogno di operare un certo numero di cicli (al fine di riscaldarsi) prima di fornire un valore attendibile; la durata di questo periodo è di circa trenta secondi.

Una volta ultimato questo processo, i dati vengono immagazzinati in memoria, e sono pronti per essere trasmessi.

E' importante ricordare che la stringa contenente i risultati dell'ultimo sensing rimarrà all'interno del sistema anche dopo la comunicazione, in modo da permetterne la visualizzazione su display. Essa verrà sovrascritta, una volta trascorsi sessanta minuti, dalla successiva misurazione.

### 3.2.3.2 Comunicazione

La fase di comunicazione ha come obiettivo finale quello di permettere all'utente di visionare i dati raccolti dalla stazione rendendoli accessibili in maniera immediata a partire dalla loro misurazione.

A tale scopo sono state studiate tre diverse modalità attraverso cui è possibile disporre delle misurazioni:

- visualizzazione tramite browser web di una pagina personale nel dominio *floweather.com*, contenente l'anagrafica di tutte le misure effettuate;
- comunicazione dell'ultima misurazione su dispositivo mobile tramite l'invio di un SMS alla stazione;
- visualizzazione diretta dell'ultima misurazione direttamente sul display della stazione.

Ogni qualvolta viene effettuata una procedura di sensing, il dispositivo provvede ad inviare l'insieme delle misure a disposizione ad un server, che a sua volta ha il compito di memorizzarle in un file XML specifico. La trasmissione è gestita direttamente dal modulo GM862 e avviene mediante protocollo GPRS. Il sistema si connette quindi ad internet dalla rete telefonica mobile attraverso un APN, contatta il server al suo indirizzo, procede ad instaurare una connessione TCP tramite l'apertura di un socket, ed infine trasmette i dati prima di terminare la stessa. Dal momento che la comunicazione avviene sulla rete mobile GSM, è necessario accertarsi che la zona nella quale si andrà a collocare la stazione sia coperta dal segnale.

Pur rappresentando indubbiamente la procedura più importante relativamente alla gestione dei dati, quella appena presentata non è tuttavia l'unica ad essere abilitata dalla piattaforma; essa, infatti, è anche in grado di rispondere a specifiche richieste inviate tramite SMS, restituendo, sempre tramite un messaggio dello stesso formato, il set contenente gli ultimi valori misurati.

Ad ogni nuova accensione, prima di iniziare la comunicazione con l'esterno, il modulo controlla lo stato della propria memoria di ricezione SMS, e, in caso di arrivo di nuovi messaggi, processa singolarmente ciascuno di essi. Ciò avviene dapprima controllando se all'interno del testo è presente una particolare stringa che è utilizzata come chiave di verifica di autenticità, quindi estraendo il numero del mittente, ed inviando infine la risposta contenente gli ultimi dati a disposizione.

Esaurite queste operazioni il messaggio è cancellato e, qualora ne siano presenti altri, si passa ad analizzare i successivi. Una volta completate le attività di gestione degli SMS e trasmissione GPRS, la stazione ha completato il suo ciclo di funzionamento e può tornare nello stato di inattività per risvegliarsi al successivo segnale di interrupt, che scatterà all'ora successiva.

La stringa contenente gli ultimi valori misurati resta tuttavia sempre in memoria; in tal modo, se durante questo periodo di spegnimento l'utente fosse interessato a conoscere queste grandezze, può utilizzare lo schermo LCD per avere una lettura immediata delle stesse.

Un'ultima caratteristica fornita dal modulo GSM è quella che permette di operare l'autolocalizzazione del dispositivo a partire dai dati di cella cui esso è collegato, i quali, una volta inviati al server, possono essere risolti tramite un'interrogazione nella latitudine e longitudine, fornendo così la posizione geografica.

### 3.2.3.3 Controllo dei consumi

Dal momento che l'intera stazione viene alimentata da una coppia di batterie LiIon ricaricabili, è evidentemente necessario cercare di ridurre al minimo i cicli di funzionamento della stessa, in modo da ottenere una maggiore durata dell'apparato riducendone, di conseguenza, le necessità di caricamento.

A tal fine sono stati individuati due possibili stati in cui il sistema può trovarsi, denominati rispettivamente 'sleep' e 'active', attraverso la gestione dei quali si è in grado di limitare i consumi energetici.

Lo stato 'sleep' sta ad indicare che la stazione è inattiva, condizione in cui essa si trova durante la larga maggioranza del tempo e che permane fino a quando il clock non alza un segnale di interrupt, segnalando in questo modo la necessità di cambiare stato. Ciò accade periodicamente ogni ora. In questa configurazione tutti i componenti risultano spenti dal momento che nessuno di essi è alimentato; soltanto il clock, grazie alla presenza della sua batteria di backup, è in grado di tenere il conteggio del tempo reale e gestire gli allarmi, mentre la scheda Arduino viene posta in modalità 'power down', in grado di ricevere unicamente segnali di interrupt per forzarne il risveglio.

Lo stato 'active' denota invece un funzionamento a pieno regime del dispositivo, il quale, come esposto dettagliatamente in precedenza, effettuerà le operazioni di sensing, eventuale gestione degli SMS e comunicazione GPRS, per poi tornare in modalità 'sleep'. Il periodo di attività ha una durata che si attesta attorno ai novanta secondi.

Appare quindi chiaro come un sistema così progettato risulti altamente efficiente; esso infatti richiede una spesa energetica contenuta per un periodo estremamente breve di tempo.

Considerando che esso lavora per novanta secondi all'interno di ogni intervallo di un'ora, questo rappresenterà soltanto il 2,5% (90s/3600s) del tempo totale, mentre il restante 97,5% non richiederà alcun consumo.

Accanto a questo lavoro di riduzione dei periodi di attività, ne è stato svolto un altro, altrettanto importante, mirato ad alleggerire al massimo il codice eseguito, diminuendo così il numero di operazioni che il sistema deve svolgere. Questa ottimizzazione, ottenuta attraverso una programmazione snella ed essenziale, punta ad alleggerire il carico di processing sul microcontrollore, garantendo prestazioni migliori sia in termini di velocità che di risparmio energetico.

### 3.2.4 Progetto

Il lavoro svolto per costruire il dispositivo è consistito principalmente in due aspetti: il primo, concernente l'hardware ed essenzialmente pratico, è consistito nella realizzazione del circuito elettrico complessivo per curare il collegamento tra i diversi componenti dello stesso; il secondo, più teorico e relativo al software, ha riguardato la scrittura del codice da caricare sulla scheda Arduino in modo da implementare lo svolgimento delle operazioni che ne caratterizzano il comportamento.

Si è ritenuto inoltre opportuno raccogliere in due distinte tabelle i dati relativi ai consumi di tutti i componenti e i loro costi, così da avere una visione più completa rispetto all'impatto che ciascuno di essi produce in termini economici ed energetici.

#### 3.2.4.1 Schema elettrico

La scheda Arduino è alimentata direttamente da una coppia di batterie che forniscono complessivamente 7,4V e 2000mAh, e svolge la funzione di microcontrollore, pilotando attraverso i numerosi pin I/O tutti gli altri componenti, e fornendogli a sua volta una carica pari a 5V. L'unico elemento che necessita di un carico minore è il modulo GM862 a cui è stato interposto un partitore di tensione in grado di fornire i 3,8V richiesti dallo stesso.

I sensori MPX4115 e TGS2442 sono connessi alla scheda per mezzo di un'interfaccia di applicazione *1-wire* di tipo analogico, mentre il restante SHT15 utilizza un protocollo *2-wire serial interface* di tipo digitale.

Il modulo GMS862, il clock BV4236 e lo schermo LCD si basano invece sul sistema di comunicazione seriale denominato *I2C (Inter Integrated Circuit)*; in particolar modo, per il collegamento del display a cristalli liquidi è stato impiegato un espansore a otto porte, il PCF8574A, che permette una facile gestione delle numerose connessioni richieste.

Il circuito di ricarica e monitoraggio della batteria è stato progettato per segnalare, tramite l'accensione di un led, quando la pila è prossima ad esaurirsi e, contemporaneamente, per permettere il ripristino della carica elettrica, proteggendo il pack di batterie da un eventuale sovraccarico.

La realizzazione dello schema elettrico è stata effettuata grazie all'impiego del software *EAGLE (Easily Applicable Graphical Layout Editor)*, sviluppato da CadSoft [19], che è risultato, per la sua semplicità, un utile strumento di rappresentazione circuitale. Esso permette altresì la progettazione PCB per la creazione di schemi per circuiti stampati, effettuata nell'ottica di uno sviluppo futuro che dovrebbe portare al passaggio dalla board di prototipo ad un circuito stampato ad hoc.

### 3.2.4.2 Codice sorgente

Il microprocessore sulla scheda è programmato utilizzando il linguaggio di programmazione Arduino (basato su Wiring) e l'ambiente di sviluppo Arduino (basato su Processing).

Della totalità del codice scritto per il dispositivo, circa 2000 righe complessivamente, la parte più rilevante è quella relativa le librerie dei diversi componenti, mentre la restante è costituita dal programma che viene caricato e fatto girare sulla scheda hardware.

Per ciascun componente è stata infatti realizzata una specifica libreria che incorpora tutte le funzioni implementate; essa è costituita da due file (H e CPP) contenenti l'insieme dei parametri, delle classi e dei metodi utilizzati.

Il programma che abilita il funzionamento della stazione, o Main, è un file in formato PDE contenente due funzioni, la prima denominata setup, la quale comprende le operazioni che vengono compiute una sola volta, all'iniziale accensione del sistema, e la seconda detta loop, che raccoglie una serie di azioni ripetute ciclicamente finché la scheda non viene riavviata o spenta.

Durante il setup il dispositivo si limita a operare un test per verificare la connettività, inviando al server i dati della cella GSM di riferimento, che verranno successivamente sfruttati per effettuare la localizzazione.

Una volta ultimato il setup, il microcontrollore entra nel loop dove, in assenza di segnali di interrupt, permane nello stato di sleep. I due eventi in grado di riattivare la stazione sono legati all'allarme orario generato dal clock e all'attivazione del display da parte dell'utente, che corrispondono rispettivamente alle due modalità di esecuzione dei comandi.

Nel caso di interrupt innescato automaticamente ogni ora il sistema svolge tre distinte funzioni: effettua il sensing, comunica i dati raccolti al server e risponde ad eventuali SMS ricevuti inviando tali misure.

Al contrario, nel caso di interrupt attivato manualmente, l'unica azione compiuta è quella di visualizzazione su schermo dei dati contenuti in memoria. Nel caso non sia presente alcuna misura, apparirà un messaggio che invita ad attendere la prossima fase di sensing.

Al fine di rendere maggiormente chiaro e leggibile il codice sviluppato, esso è stato completamente integrato da una serie di commenti supplementari realizzati con *Doxygen* [20], un'applicazione per la generazione automatica della documentazione a partire dal codice sorgente di un generico software. Doxygen è un progetto open source, rilasciato sotto licenza GPL.

```
/*  
MAIN PROGRAM V_2.1 (jun 2010)
```

```
AUTHORS: Stefano Manni, Domenico Schillaci
```

```
FEATURES:
```

```
* SENSING  
* TEMPORIZATION (SLEEP)  
* VISUALIZATION (LCD)  
* COMMUNICATION (xml file)  
* SMS REPLY
```

```
WHAT'S NEW:
```

```
* Visualization on lcd display  
*****/
```

```
/*  
* INCLUDE LIBRARIES *
```

```
*****/  
#include <BV4236.h>  
#include <MPX4115.h>  
#include <SHT15.h>  
#include <TGS2442.h>  
#include <GM862.h>  
#include <LCDI2C.h>  
#include <NewSoftSerial.h>  
#include <Wire.h>  
#include <avr/sleep.h>  
#include <math.h>
```

```
/*  
* PINS MACROS *
```

```
*****/  
//SHT15 - Temperature and humidity sensor  
#define SHT15_DATA_PIN 4  
#define SHT15_CLOCK_PIN 5  
//MPX4115 - Pressure sensor  
#define MPX4115_DATA_PIN 0 // analog  
//TGS2442 - Carbon monoxide sensor  
#define TGS2442_HEATER_PIN 6  
#define TGS2442_CIRCUIT_PIN 7  
#define TGS2442_SENSING_PIN 1 // analog  
//BV4236 - Real time clock  
#define BV4236_INTERRUPT_PIN 1 // pin 3  
//GM862 - GSM/GPRS Module  
#define GM862_ONOFF_PIN 10  
#define GM862_RX_PIN 8  
#define GM862_TX_PIN 9  
//LCDI2C - Display  
#define LCDI2C_N_ROWS 4  
#define LCDI2C_N_COLS 20  
//OTHERS  
#define PUSH_INTERRUPT_PIN 0 // pin 2  
#define POWER_ON_LED 13
```

```

/*****
*                               MACROS                               *
*****/
//sensing
#define N_SAMPLES      21          //n of samples for tem, hum and prs
#define T_SAMPLES      500        //interval btw 2 samples tem, hum, prs
#define COX_N_SAMPLES  40         //interval btw 2 samples fixed at 1s
//communication
#define PIN             "2010"     //Personal Identifier Number
#define SID             "3298503095" //Station Identifier
#define MNC             "88"      //Mobile Network Code (Wind)
#define MCC             "222"     //Mobile Country Code (It)

/*****
*                               CONSTRUCTORS                          *
*****/
//sensing
MPX4115 MPX4115 (MPX4115_DATA_PIN);
SHT15 SHT15 (SHT15_DATA_PIN, SHT15_CLOCK_PIN);
TGS2442 TGS2442 (TGS2442_HEATER_PIN, TGS2442_CIRCUIT_PIN, TGS2442_SENSING_PIN);
//sincronization
BV4236 BV4236;
//communication
NewSoftSerial serial(GM862_RX_PIN, GM862_TX_PIN);
GM862 GM862(&serial, GM862_ONOFF_PIN);
//visualization
LCDI2C LCDI2C (LCDI2C_N_COLS, LCDI2C_N_ROWS);

/*****
*                               VARIABLES                              *
*****/
//mode=0 sensing/communication | mode=1 visualization
int mode=1;

//setup variables
boolean flagReset=true;          //a reset or a power down was happened
boolean flagInterrupt=false;    //queue rtc interrupt during visualization

//float sensing variables
float temF;
float prsF;
float humF;
float coxF;

//string sensing variables
char tem[8];
char hum[8];
char prs[8];
char cox[8];
char data[70];                  //all measurements or lac/cid concatenated

//location variables
char lac[7];                    //Location Are Code
char cid[7];                     //Cell ID

```

```

/*****
*                               SETUP                               *
*****/
void setup(){

    Serial.begin(57600);
    digitalWrite(POWER_ON_LED,HIGH);

    //switch on and initialize gm862 modem
    GM862.switchOn();
    GM862.init(PIN);

    //print welcome message
    LCIDI2C.switchOn();
    LCIDI2C.init();                               //initialize the LCIDI2C
    LCIDI2C.print("_ _::floweather::_ _");
    LCIDI2C.setCursor(1,6);
    LCIDI2C.print("Welcome!");
    LCIDI2C.setCursor(2,4);
    LCIDI2C.print("Localization,");
    LCIDI2C.setCursor(3,0);
    LCIDI2C.print("connection_ _testing..");
    myDelay(5000);
    LCIDI2C.switchOff();

    //check if is registered, get data cell and send it to server
    while (!GM862.isRegistered()) {
        GM862.initGPRS();                          //setup of GPRS context
        do
        {
            GM862.enableGPRS();
            myDelay(50);
        }
        while (!GM862.GPRSIsActivated());

        GM862.getDataCell(lac, cid);               //get data cell
        strcpy(data, "sid=");
        strcat(data, SID);
        strcat(data, "&mnc=");
        strcat(data, MNC);
        strcat(data, "&mcc=");
        strcat(data, MCC);
        strcat(data, "&lac=");
        strcat(data, lac);
        strcat(data, "&cid=");
        strcat(data, cid);

        while (!GM862.openSocket()){} //open a socket
        while (!GM862.writeAndReadSocket(data)){} //send data to server

        GM862.disableGPRS();
        GM862.switchOff();
        BV4236.resetAlarm();
        attachInterrupt(BV4236_INTERRUPT_PIN,handlerRtc , FALLING);
}

```

```

/*****
*
*                               LOOP
*
*****/
void loop(){

//SENSING AND COMMUNICATION MODE
if (mode==0){
    detachInterrupt(PUSH_INTERRUPT_PIN); //detach push interrupt
    digitalWrite(POWER_ON_LED,HIGH);

    //switch on and initialize gm862 modem
    GM862.switchOn();
    GM862.init(PIN);

    /***** SENSING *****/
    flagInterrupt=false;
    flagReset=false;

    //compute median
    float    temArray[N_SAMPLES];
    float    humArray[N_SAMPLES];
    float    prsArray[N_SAMPLES];
    int i=0;                                     //just a counter

    //catch samples
    while (i<N_SAMPLES){
        temArray[i]=SHT15.getTem();
        humArray[i]=SHT15.getHum();
        prsArray[i]=MPX4115.getPrs();
        i++;
        myDelay(T_SAMPLES);
    }

    bubbleSort(temArray, N_SAMPLES);
    bubbleSort(humArray, N_SAMPLES);
    bubbleSort(prsArray, N_SAMPLES);
    temF = temArray[(N_SAMPLES-1)/2];
    humF = humArray[(N_SAMPLES-1)/2];
    prsF = prsArray[(N_SAMPLES-1)/2];
    coxF = TGS2442.getCox(temF, COX_N_SAMPLES);

    floatToString(temF, tem);
    floatToString(humF, hum);
    floatToString(prsF, prs);
    floatToString(coxF, cox);
    strcpy(data, "sid=");
    strcat(data, SID);
    strcat(data, "&tem=");
    strcat(data, tem);
    strcat(data, "&hum=");
    strcat(data, hum);
    strcat(data, "&prs=");
    strcat(data, prs);
    strcat(data, "&cox=");
    strcat(data, cox);

```

```

/*****
*                               LOOP                               *
*****/

/***** COMMUNICATION *****/
//check if is registered, get data cell and send it to server
while (!GM862.isRegistered()) {}

GM862.initGPRS();           //setup of GPRS context
do
{
    GM862.enableGPRS();
    myDelay(50);
}

while (!GM862.GPRSIsActivated());

while (!GM862.openSocket() ){} //open a socket

while (!GM862.writeAndReadSocket(data)){}

GM862.disableGPRS();

/***** SMS *****/
GM862.initSMS();

char body[15];
char sender[15];

int nUnread=GM862.getNUnreadSMS();

for (i=1;i<=nUnread;i++) {
    GM862.readSMS(i,body, sender);

    if (strstr(body,PIN)){
        strcpy(data, "floweather: Tem=");
        strcat(data, tem);
        strcat(data, "C, Prs=");
        strcat(data, prs);
        strcat(data, "hPa, Hum=");
        strcat(data, hum);
        strcat(data, "%, CO=");
        strcat(data, cox);
        strcat(data, "ppm");

        GM862.sendSMS(sender, data);
    }
    GM862.deleteSMS(i);
}

GM862.switchOff();
}
//END SENSING MODE

```

```

/*****
*                                     LOOP                                     *
*****/

//VISUALIZATION MODE
else if (mode==1){
    detachInterrupt(PUSH_INTERRUPT_PIN); //detach pushbutton interrupt
    digitalWrite(POWER_ON_LED,HIGH);

    if(flagReset==true){           //a reset or a power down has happened
        //print welcome message
        LCDI2C.switchOn();
        LCDI2C.init();           //initialize the LCDI2C
        LCDI2C.print("_ _::floweather::");
        LCDI2C.setCursor(1,1);
        LCDI2C.print(".. is _now_ working,");
        LCDI2C.setCursor(2,3);
        LCDI2C.print("wait _until_ the");
        LCDI2C.setCursor(3,2);
        LCDI2C.print("next _measurement");
        myDelay(5000);
        LCDI2C.switchOff();
    }

    else if(flagReset==false){           //at least one measurement
        //print data and time
        int    Minute;           //store minute value
        int    Hour;           //store hour value
        int    Day;           //store day value
        int    Date;           //store date value
        int    Month;           //store month value
        int    Year;           //store year value

        BV4236.getTime(&Year, &Month, &Date, &Day, &Hour, &Minute);

        LCDI2C.switchOn();
        LCDI2C.init();           //initialize the LCDI2C
        //title and data
        LCDI2C.print("_ _::floweather::");
        LCDI2C.setCursor(1,0);
        LCDI2C.print( "_");
        LCDI2C.print(getDayOfWeek(Day));
        LCDI2C.print("_");
        LCDI2C.print((Date >> 4) +48);
        LCDI2C.print((Date & 0xF)+48);
        LCDI2C.print("/");
        LCDI2C.print((Month >> 4) +48);
        LCDI2C.print((Month & 0xF)+48);
        LCDI2C.print("/20");
        LCDI2C.print((Year >> 4) +48);
        LCDI2C.print((Year & 0xF)+48);
        LCDI2C.print("_h");
        LCDI2C.print((Hour >> 4) +48);
        LCDI2C.print((Hour & 0xF)+48);
    }
}

```

```

/*****
*                                     LOOP                                     *
*****/

        //measurements
        LCDI2C.setCursor(2,0);
        LCDI2C.print("T=");
        LCDI2C.print(tem);
        LCDI2C.print("C□");
        LCDI2C.setCursor(2,8);
        LCDI2C.print("□P=");
        LCDI2C.print(prs);
        LCDI2C.print("hPa");
        LCDI2C.setCursor(3,0);
        LCDI2C.print("H=");
        LCDI2C.print(hum);
        LCDI2C.print("%");
        LCDI2C.setCursor(3,9);
        LCDI2C.print("CO=");
        LCDI2C.print(cox);
        LCDI2C.print("ppm");
        myDelay(5000);

        LCDI2C.switchOff();
    }
}
//END VISUALIZATION MODE

//SLEEPING
if (flagInterrupt==false){
    digitalWrite(POWER_ON_LED,LOW);
    sleepNow();
}
//END SLEEPING
}

/*****
*                                     SLEEP NOW                                     *
*****/
void sleepNow(){

    set_sleep_mode(SLEEP_MODE_PWR_DOWN);    //sleep mode is set here

    sleep_enable();        //enables the sleep bit in the mcu register

    attachInterrupt(PUSH_INTERRUPT_PIN, handlerPush, FALLING);

    BV4236.resetAlarm();
    sleep_mode();

    sleep_disable();
}

```

### 3.2.4.3 Tabella dei consumi

Nello stilare una tabella riassuntiva dei consumi di ciascun componente, si è reso necessario operare una distinzione tra i due diversi modi di funzionamento descritti in precedenza, ovvero gli stati 'active' e 'sleep'.

A seconda dello stato in cui si trova il dispositivo, infatti, vi sarà una corrispondente variazione dei costi energetici.

Nel primo caso i componenti sono tutti in funzione eccetto lo schermo la cui accensione viene inibita durante il processo di sensing e trasmissione dei dati; i consumi maggiori sono quelli prodotti dal modulo GSM, indubbiamente il più complesso tra gli elementi assemblati, il quale necessita anche durante la fase di accensione di un picco di corrente di circa due Ampere. Le stime sul consumo dei componenti sono state effettuate per eccesso, considerando ad esempio per i sensori una corrente di alimentazione pari a quella richiesta in fase di misurazione, sebbene dei circa novanta secondi che costituiscono il ciclo di accensione periodico del dispositivo, solo una frazione di essi sono realmente utilizzati per le misure.

Al contrario, quando il sistema entra in sleep mode la linea elettrica responsabile dell'alimentazione dei sensori viene cortocircuitata grazie all'impiego di un transistor; stessa cosa avviene per il clock, il quale passa in modalità di backup mantenendo il solo oscillatore in funzione grazie alla propria batteria. Anche il modulo telefonico non incide sui consumi, poiché si è provveduto a spegnerlo al termine di ogni fase di comunicazione. La scheda Arduino, invece, possiede sei diverse modalità a basso consumo impostabili; nel caso specifico è stata scelta quella che produce i migliori vantaggi in termini di risparmio energetico. La modalità 'power down', infatti, abilita unicamente la ricezione dei segnali di interrupt, inibendo ogni altro tipo di ingresso o uscita.

Occorre infine considerare, l'impatto prodotto dall'accensione dello schermo LCD durante la fase di sleep, il quale, pur non avendo valori particolarmente elevati (grazie anche alla scelta di un timer di cinque secondi che regola la visualizzazione su display), può incidere se attivato più volte durante il giorno. Nello stilare la tabella, è stata effettuata una stima che prevede, in media, una accensione all'ora.

Componente	ACTIVE	MODE	SLEEP	MODE
	consumo	stato	consumo	stato
Scheda Arduino 2009	25mA	ON	1mA	PWR DOWN
Sensore SHT15	0.6mA	MSR	0mA	OFF
Sensore MPX4115	7mA	MSR	0mA	OFF
Sensore TGS2442	2.8mA	MSR	0mA	OFF
Real Time Clock BV4236	0.4mA	ON	0mA	BACKUP
Modulo GM862 QUAD	20mA	ON	0mA	OFF
Dicplay LCD HD44780	0mA	OFF	0,35mA	5sec ON
Totale	55.8mA		1,35mA	

Tabella 3.2: Componenti e relativi consumi a seconda dei modi di funzionamento

### 3.2.4.4 Tabella dei costi

In questa tabella sono stati riassunti i costi relativi ai diversi componenti che costituiscono il dispositivo.

Occorre inoltre ricordare che poiché si utilizza una connessione alla rete telefonica mobile, ogni trasmissione dati avrà un certo costo. Ciò porta alla necessità di inserire nel dispositivo una scheda SIM che sia sempre provvista di credito.

La scelta del provider di telefonia mobile a cui affidarsi, tra i numerosi disponibili, è ricaduta infine su WIND, in quanto capace di proporre le tariffe più convenienti. Il costo di una connessione ad internet tramite GPRS è infatti di 0,3 centesimi di euro per kB (inviato o ricevuto), mentre l'invio di un SMS ha un prezzo di 9 centesimi.

Non sono stati presi in considerazione i costi relativi alla produzione dell'involucro rappresentato dal vaso, poiché trattandosi di una lavorazione tipicamente industriale, il prezzo di un singolo esemplare sarebbe risultato eccessivamente elevato e quindi non in grado di fornire un riferimento attendibile.

Sono stati infine tralasciati i costi relativi al materiale elettrico complementare, come fili, resistenze e transistor, dal momento che hanno complessivamente un impatto minimo rispetto ai componenti, e possono di conseguenza essere trascurati.

Il costo totale raggiunto è da considerarsi puramente di riferimento in quanto, in caso di produzione in serie di numerosi esemplari, i singoli elementi potranno essere acquistati ad un prezzo ben inferiore rispetto a quello sostenuto in questa circostanza.

Componente	Prezzo
Scheda Arduino 2009	€ 25.20
Sensore SHT15	€ 28.00
Sensore MPX4115	€ 10.90
Sensore TGS2442	€ 13.20
Real Time Clock BV4236	€ 15.00
Modulo GM862 QUAD	€ 24.90
Board 50pin GM862	€ 20.00
Antenna PCB Mini 34x7mm	€ 8.60
SIM Card	€ 10.00
Display LCD HD44780	€ 5.50
Espansore PCF8574A	€ 3.30
LIon Battery Pack	€ 27.20
1.3W 10V Solar Cell	€ 20.60
Totale	€ 191.80

Tabella 3.3: Componenti e relativi costi

### 3.2.5 Componenti aggiuntivi

Come indicato in precedenza, l'importante caratteristica di modularità rende il progetto espandibile in qualsiasi momento, tramite l'aggiunta di moduli supplementari, con l'obiettivo di abilitare nuove funzionalità rispetto all'implementazione di board di sensori per misure integrative o di nuovi protocolli per la comunicazione.

In sede di progetto, sono state effettuate diverse ipotesi riguardo alcuni tra i possibili componenti che, affiancati al dispositivo, potrebbero contribuire ad accrescerne l'utilità.

Tra questi sono stati selezionati i cinque che sono stati ritenuti maggiormente adatti e sviluppabili nel breve periodo.

#### 3.2.5.1 Pannello solare

Non rappresenta un modulo assestante in senso stretto, poiché è stato inserito direttamente nel prototipo di base, tuttavia l'impiego di una cella solare non appare indispensabile per il funzionamento del dispositivo, in quanto le batterie possono essere ricaricate anche con un semplice trasformatore. L'alimentazione tramite pannello solare permette d'altra parte di rendere la stazione completamente autosufficiente, e in grado di ricaricarsi da sola nel tempo senza bisogno di essere posizionata vicino ad una presa elettrica.

#### 3.2.5.2 Anemometro

Per rendere più complete le informazioni meteorologiche raccolte è essenziale aggiungere a quelle già a disposizione delle altre, concernenti la velocità e la direzione del vento. L'impiego di un anemometro appare di conseguenza di grande importanza, ma per problemi legati al costo, alla complessità e alle notevoli dimensioni non è stato possibile inserirlo all'interno del modulo base.

La progettazione 'in house' di un modulo costituito da un anemometro studiato specificatamente per integrarsi alla stazione *floweatherpot* rappresenta indubbiamente una tra le prossime sfide da raccogliere.

#### 3.2.5.3 Board di sensori per la qualità dell'aria

Il sensore in grado di rilevare la concentrazione di monossido di carbonio presente nell'ambiente serve a dare un parametro di riferimento generico in merito alla salubrità dell'aria; tuttavia, per effettuare un'analisi più precisa occorre dotarsi di una piccola piattaforma che integri sensori specifici per la misurazione di gas nocivi quali anidride carbonica (CO<sub>2</sub>), biossido di azoto (NO<sub>2</sub>), biossido di zolfo (SO<sub>2</sub>) e benzene (C<sub>6</sub>H<sub>6</sub>). Essi costituiscono assieme al CO i principali componenti responsabili dell'inquinamento atmosferico.

Tale espansione permetterebbe ad esempio di operare un'analisi molto più approfondita sulla qualità dell'aria respirata quotidianamente dai cittadini nei centri urbani.

#### **3.2.5.4 Board di sensori per l'inquinamento acustico**

Un altro importante aspetto potrebbe essere quello di monitorare altre tipologie di inquinamento oltre a quello atmosferico, così da operare un'indagine sulla qualità dell'ambiente circostante a trecentosessanta gradi, prendendo in considerazione tutti i diversi fattori che lo influenzano.

Da questo punto vista appare interessante uno studio volto a misurare il grado d'inquinamento acustico presente sul territorio, rispetto al quale ad oggi non viene effettuato alcun controllo, sebbene alti livelli dello stesso possano arrecare, se mantenuti in maniera costante, seri disturbi alla salute delle persone.

Tramite l'impiego di un piccolo microfono è possibile misurare la portata dei decibel prodotti dall'ambiente e studiarne l'evoluzione nel tempo o nello spazio combinando insieme i valori recuperati da diversi dispositivi.

#### **3.2.5.5 Modulo di comunicazione basato sul protocollo ZigBee**

La scelta di impiegare il protocollo GSM/GPRS come standard per la gestione della comunicazione, per quanto ritenuta valida, non rappresenta tuttavia l'unica soluzione implementabile per un sistema con le caratteristiche di FLOWEATHER. La necessità di inserire una carta SIM per abilitare il funzionamento del dispositivo, oltre al costo per ogni SMS inviato o per ogni una connessione GPRS effettuata rappresentano senza alcun dubbio una limitazione non trascurabile. A tal fine sono state pensate altre possibilità basate sull'impiego di nuovi protocolli di comunicazione che permettano di superare queste restrizioni, pur avendo a loro volta degli svantaggi. Tra queste, quella che appare più interessante consiste nella creazione di un modulo aggiuntivo basato sullo standard 802.15.4, che fornisce connettività radio a corto raggio e a bassa potenza. Il protocollo in questione è denominato ZigBee e si basa su una tecnologia semplice ed economica.

Questo tipo di protocolli è progettato per abilitare la connessione tra applicazioni embedded che richiedono basso transfer rate e bassi consumi.

Per implementare una connessione tra due sistemi sono necessari due dispositivi ZigBee, uno ZigBee Coordinator (ZC), in grado di memorizzare informazioni provenienti da altri nodi ed agire come gateway inoltrandole verso un'altra rete, ed uno ZigBee End Device (ZED), che svolge l'unica funzione di comunicare con il suo nodo coordinatore. Realizzando quindi la configurazione seguente, collegando il nodo ZED alla stazione FLOWEATHER ed il nodo ZC ad un personal computer o ad un access point WiFi, si riuscirebbe così a trasmettere di volta in volta i dati acquisiti al server di destinazione. Naturalmente, per permettere questo tipo di comunicazione, il dispositivo a cui è collegato il nodo coordinatore deve essere costantemente acceso e collegato alla rete internet.

### 3.2.6 Design

L'orientamento seguito durante la fase di progettazione di *floweatherpot* è stato fortemente influenzato dalla ricerca di una forma attraverso cui rappresentare l'oggetto che risultasse originale e che permettesse al contempo di attribuire allo stesso una valenza alternativa, al fine di invogliarne gli utenti all'acquisto.

Trattandosi di un dispositivo studiato per essere posizionato prettamente all'esterno, dal momento che deve trovarsi costantemente immerso nell'ambiente da controllare, si è ritenuto opportuno trovare una soluzione che ne giustificasse la collocazione e ne permettesse l'impiego nello specifico contesto.

A partire da queste motivazioni si è giunti all'idea di realizzare un vaso da fiori, che rappresenta il contenitore del dispositivo e che può essere sistemato su un qualunque balcone o davanzale.

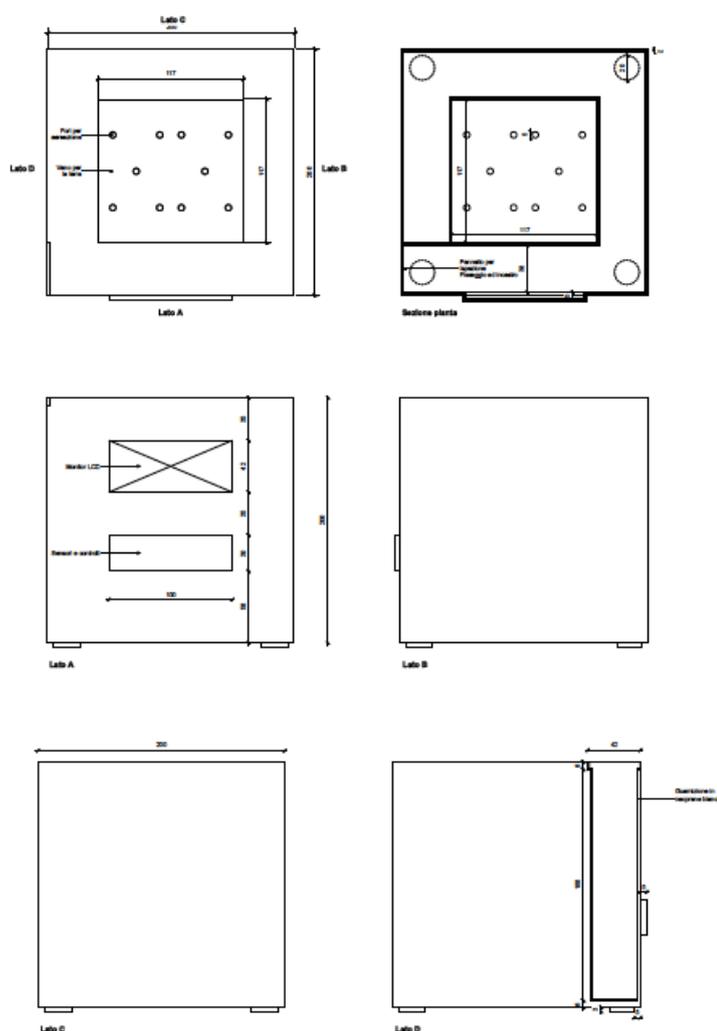


Figura 3.9: Schema di progetto

### 3.2.6.1 Materiale

La scelta del polietilene come materiale per la realizzazione del prototipo, è dovuta principalmente ai numerosi vantaggi che l'utilizzo di questa resina plastica comporta dal punto di vista ecologico. Essi sono legati al risparmio nel consumo delle materie prime, nella produzione e nel trasporto.

Nella lavorazione del polietilene, infatti, il petrolio si trasforma in materiale resistente e più volte riciclabile, a differenza di quanto avviene per il suo impiego a livello di riscaldamento e combustibile, dove si esaurisce completamente in un'unica combustione senza opportunità di riciclo. Anche in termini di effettiva produzione e trasformazione, il polietilene gode, dal punto di vista ambientale, di una posizione di favore rispetto ad altri materiali. Per effetto della bassa temperatura di lavorazione, che si aggira sui 200-250° C, esso necessita, rispetto ad altri materiali come ad esempio il vetro (temperatura di lavorazione da 500 a 600° C) o l'acciaio (da 800 a 1000° C), di un quantitativo di gran lunga inferiore di combustibile.

Le materie plastiche in genere, inoltre, contribuiscono in maniera ottimale a soddisfare la crescente esigenza di materiali da imballo sempre più leggeri. Con il polietilene, ad esempio, il rapporto tra il peso dell'imballo e quello del contenuto è di un ventesimo, mentre nel caso del vetro non raggiunge nemmeno ad un mezzo. L'impiego di materie plastiche consente, quindi, non solo di ridurre il peso proprio del prodotto ma anche di diminuire sensibilmente il numero di viaggi che gli autocarri devono effettuare per la distribuzione dei prodotti stessi.

### 3.2.6.2 Progetto

Per ciò che concerne la progettazione del vaso si è cercato di realizzare un oggetto dalle forme semplici ed essenziali, che seguisse, attraverso l'eleganza delle linee, il design dei moderni prodotti d'arredamento per esterni.

A tale scopo si è optato per un'estetica ispirata alla forma geometrica più elementare, quella cubica. *Flowerpot* è infatti costituito da un piccolo cubo di colore bianco, contenente al suo interno un altro cubetto che funge da recipiente per la terra. Ha dimensioni compatte (il lato esterno è lungo venti centimetri), un peso minimo ed è quindi facilmente posizionabile in spazi anche ridotti.

Su una delle quattro facce orizzontali è presente una apertura per permettere ai sensori di stare direttamente al contatto con l'ambiente; sempre sullo stesso lato, ma superiormente, è situato lo schermo LCD ed un elementare pannello di controllo da cui è possibile attivare il display. Sono inoltre presenti due led che hanno il compito di indicare rispettivamente quando la stazione è in funzione e quando la batteria deve essere ricaricata, e un connettore a cui collegarsi per fornire l'alimentazione. La totalità della componentistica elettronica è situata dentro uno spazio cavo che si trova tra la parete del contenitore e quella più interna del vaso vero e proprio; esso è reso accessibile attraverso un pannello a scorrimento situato lateralmente a tale apertura.

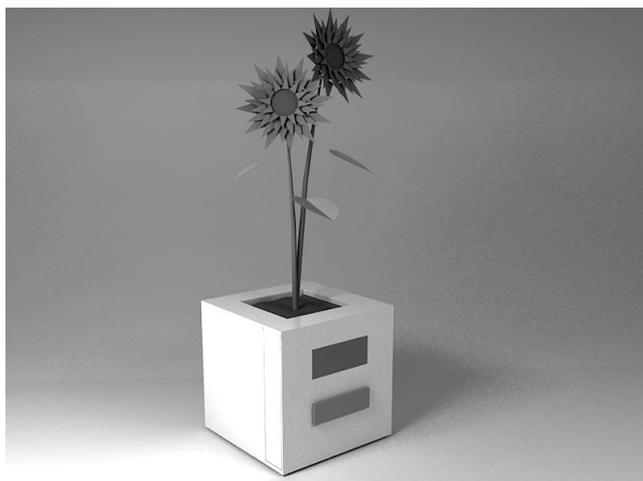


Figura 3.10: Prospettiva frontale

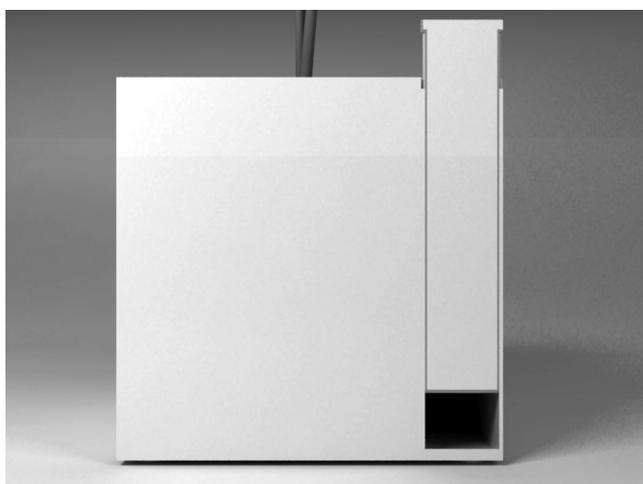


Figura 3.11: Prospettiva laterale destra

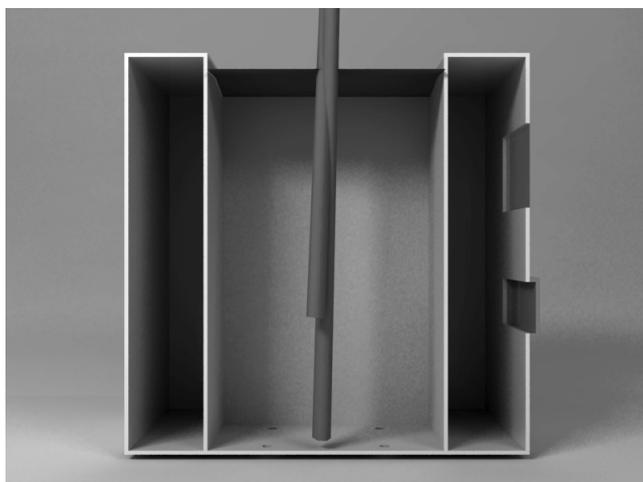


Figura 3.12: Sezione laterale

## 3.3 La pagina web: *floweather.com*

Come descritto in precedenza, le misurazioni, una volta che sono state correttamente rilevate e trasmesse dalle stazioni, vengono immagazzinate all'interno di un server. Da questo momento in poi la gestione, la manipolazione e la presentazione finale dei dati avvengono attraverso il dominio *floweather.com*, e sono realizzate grazie ad una serie di script PHP e Java.

Per ottenere questi risultati ci si è perciò indirizzati verso quelle soluzioni mirate a garantire la sicurezza e il mantenimento dell'integrità dei dati, oltre a permetterne la visualizzazione in un modo rapido e intuitivo. Anche in questa circostanza si è prestata molta attenzione al design dell'interfaccia d'utente, così da renderla semplice e originale.

### 3.3.1 Caratteristiche

Così come avvenuto per il dispositivo, si è ritenuto importante individuare quelle caratteristiche che risultassero maggiormente distintive per l'infrastruttura informatica; ovvero:

- Semplicità;
- Personalizzazione;
- Aggiornamento in tempo reale;
- Automatizzazione.

Ognuna di queste quattro proprietà contribuisce in maniera determinante a creare un servizio di presentazione dei dati che risulti ottimale.

#### 3.3.1.1 Semplicità

Il primo aspetto che è apparso importante curare è quello riguardante la strutturazione dei numerosi dati a disposizione. Essa, infatti, è realizzata in modo da risultare molto semplice, così da permettere una chiara e rapida consultazione da parte degli utenti.

Tale obiettivo è raggiunto grazie all'impiego di diversi grafici che hanno lo scopo di raggruppare le varie tipologie di misurazione per mostrarne l'evoluzione nel tempo sia dal punto di vista giornaliero che mensile. Accanto ai grafici vengono forniti, a partire dai dati a disposizione, alcuni consigli utili a comprendere meglio i mutamenti del climatici in atto, o volti a migliorare le condizioni di comfort ambientale.

Attraverso questo tipo di approccio è quindi possibile avere un'immediata idea del comportamento dell'ambiente senza bisogno di dover interpretare direttamente una grande mole di dati, la cui analisi risulterebbe altrimenti molto complessa.

### **3.3.1.2 Personalizzazione**

Alla base del successo di un qualsiasi tipo di servizio vi è, nella larga maggioranza dei casi, la capacità di personalizzazione dello stesso. Una segmentazione più fine degli utenti, consente, infatti, l'erogazione di un servizio che sia estremamente specifico e, di conseguenza, in grado di venire puntualmente incontro alle esigenze di ogni singolo fruitore.

Da questo punto di vista non fa eccezione la funzione di visualizzazione dei dati offerta da FLOWEATHER, che permette a ciascun possessore di una stazione di registrarla gratuitamente sul dominio *floweather.com*. Una volta completata la procedura di inserimento del proprio dispositivo sarà possibile accedere ad una pagina personale all'interno della quale si è in grado di navigare attraverso varie schermate contenenti i grafici, la mappa di localizzazione ed altre informazioni relative alle proprie misurazioni.

### **3.3.1.3 Aggiornamento in tempo reale**

Un'altra importante proprietà è quella che consente di avere a disposizione i dati immediatamente dopo che sono stati rilevati dal dispositivo.

Per questa ragione il sistema di memorizzazione è stato programmato in modo da mettere a disposizione l'ultima misurazione effettuata nell'istante stesso in cui essa è immagazzinata nel server; essa viene inserita nella pagina personale ed è quindi visionabile pochi secondi dopo essere stata rilevata.

### **3.3.1.4 Automatizzazione**

Per permettere un funzionamento che risulti ottimale, l'infrastruttura informatica che si occupa della gestione dei dati è stata studiata per possedere un alto grado di automatizzazione, ed essere di conseguenza in grado di coordinare diverse stazioni. Le attività di creazione, aggiornamento e modifica dei file relativi ad ogni dispositivo sono infatti effettuate in maniera del tutto automatica dagli script implementati, senza bisogno di alcun controllo da parte degli amministratori.

## **3.3.2 Funzionalità**

Il server, situato nel dominio *floweather.com*, contiene la totalità dei file che si occupano della gestione dei dati, oltre a un database con l'elenco degli utenti registrati al servizio con i relativi identificativi delle stazioni.

Le misurazioni trasmesse con cadenza oraria dai numerosi dispositivi seguono sempre lo stesso iter: dapprima sono immagazzinate all'interno di un documento specifico per ogni stazione, successivamente vengono processate al fine di calcolare o aggiornare, a seconda dei casi, campi aggregati quali massimi, minimi e medie; infine sono pronti ad essere visualizzati sotto forma di grafici contenuti nelle singole pagine personali.

Esistono inoltre altre due funzioni supplementari che consentono di utilizzare in maniera differente l'anagrafica dei dati raccolti; nel primo caso è possibile collegare al servizio un account Twitter, in modo da condividere le proprie informazioni orarie e renderle visibili tramite il popolare portale di microblogging. Nel secondo caso, si può scaricare direttamente dalla propria pagina un documento in formato EPW, che permette di utilizzare i dati climatici complessivi raccolti all'interno di specifici software per svolgere simulazioni energetiche.

### 3.3.2.1 Memorizzazione

Il modulo GM862, contenuto in ogni dispositivo, permette l'invio tramite GPRS delle misurazioni raccolte durante la fase di sensing.

I dati ricevuti dal server, vengono così immagazzinati in un file XML che prende il nome dal numero telefonico della SIM utilizzata per la trasmissione, in modo da identificare univocamente ogni stazione. Ad ogni comunicazione in entrata, il sistema controlla se esiste già un documento con l'identificativo in questione e, in caso negativo, provvede a crearlo.

Il file in questione possiede una struttura ad albero, che si sviluppa a partire da due campi: il primo, denominato 'geo', contiene le informazioni relative alla localizzazione del dispositivo; il secondo, definito 'data', raccoglie le diverse misurazioni registrate nel tempo. A partire dai dati di cella inviati dal modulo (MCC, MNC, CID e LAC) è possibile risalire alle coordinate di latitudine e longitudine relative alla Base Station cui il dispositivo è connesso, che a loro volta possono essere risolte nell'indirizzo. Questo tipo di localizzazione, essendo basato sulla cella di appartenenza, fornisce una precisione minore rispetto a quella garantita da sistemi specifici per il rilevamento della posizione come il GPS, tuttavia, non necessita l'implementazione di nuovi componenti (che si rifletterebbe anche sul prezzo finale), fornendo un'accuratezza media in un raggio di circa trecento metri. Un altro dato che viene calcolato è quello relativo all'altitudine, che risulta necessario per riportare la pressione atmosferica misurata dal valore assoluto a quello relativo (pressione atmosferica a livello del mare), generalmente usato come riferimento in meteorologia. Il campo aggregato 'data' si occupa invece di immagazzinare nel tempo le misure ricevute; ognuna di esse è identificata da una diversa 'entry', caratterizzata da cinque attributi che indicano la data (anno, mese, giorno, ore e minuti), e contenente il vettore con i quattro tipi di misurazioni (temperatura, umidità, pressione e monossido di carbonio). L'inserimento delle entry è stato studiato in modo da porre la più recente in cima alla lista, e di seguito le altre.

A partire da questo file, ne viene inoltre creato un secondo, definito 'extended', che si occupa di calcolare i valori massimi, minimi e medi per ogni giorno ed ogni mese dell'anno. Esso segue la stessa configurazione del primo e contiene due campi: 'days' e 'months'. Il primo calcola i tre parametri in ambito giornaliero a partire dalle ventiquattro misurazioni orarie, il secondo effettua lo stesso procedimento ma rispetto ai giorni in un mese. In questo modo i massimi, i minimi e le medie vengono costantemente aggiornati ad ogni nuova entry, e sono disponibili in tempo reale; in aggiunta il server non viene sovraccaricato poiché distribuisce nel tempo il numero di operazioni da svolgere.

//Esempio di file XML BASE

```
<xml>
  <geo>
    <lat>45.476894</lat>
    <lng>9.223275</lng>
    <alt>109</alt>
    <add>Viale Romagna, 49-57, 20133 Milan, Italy</add>
    <date>Thu, 21 Jul 2010, 11:31</date>
  </geo>
  <data>
    <entry y="2010" m="07" d="21" h="14" ms="01">
      <tem>27.6</tem>
      <hum>47.5</hum>
      <prs>1013</prs>
      <cox>0.3</cox>
    </entry>
    <entry y="2010" m="07" d="21" h="13" ms="01">
      <tem>27.3</tem>
      <hum>47.1</hum>
      <prs>1014</prs>
      <cox>0.2</cox>
    </entry>
    <entry y="2010" m="07" d="21" h="12" ms="01">
      <tem>27.2</tem>
      <hum>46.9</hum>
      <prs>1012</prs>
      <cox>0.3</cox>
    </entry>
    <entry y="2010" m="07" d="21" h="11" ms="01">
      <tem>27.0</tem>
      <hum>47.1</hum>
      <prs>1010</prs>
      <cox>0.1</cox>
    </entry>
    <entry y="2010" m="07" d="21" h="10" ms="01">
      <tem>26.8</tem>
      <hum>46.9</hum>
      <prs>1011</prs>
      <cox>0.1</cox>
    </entry>
    <entry y="2010" m="07" d="21" h="09" ms="01">
      <tem>27.0</tem>
      <hum>47.1</hum>
      <prs>1010</prs>
      <cox>0.1</cox>
    </entry>
    <entry y="2010" m="07" d="21" h="08" ms="01">
      <tem>26.8</tem>
      <hum>46.9</hum>
      <prs>1011</prs>
      <cox>0.2</cox>
    </entry>
    <...>
  </data>
</xml>
```

//Esempio di file XML EXTENDED

```
<xml>
  <days>
    <entry y="2010" m="07" d="21" c="10">
      <tem>
        <max>28.3</max>
        <min>26.5</min>
        <avg>27.42</avg>
      </tem>
      <hum>
        <max>60.9</max>
        <min>46.9</min>
        <avg>51.44</avg>
      </hum>
      <prs>
        <max>1020</max>
        <min>1006</min>
        <avg>1012</avg>
      </prs>
      <cox>
        <max>3.5</max>
        <min>0.1</min>
        <avg>0.3</avg>
      </cox>
    </entry>
    <...>
  </days>
  <months>
    <entry y="2010" m="07" c="108">
      <tem>
        <max>28.3</max>
        <min>26.5</min>
        <avg>27.42</avg>
      </tem>
      <hum>
        <max>60.9</max>
        <min>46.9</min>
        <avg>51.44</avg>
      </hum>
      <prs>
        <max>1021</max>
        <min>1002</min>
        <avg>1013</avg>
      </prs>
      <cox>
        <max>8.3</max>
        <min>0.1</min>
        <avg>0.3</avg>
      </cox>
    </entry>
    <...>
  </months>
</xml>
```

### 3.3.2.2 Presentazione

La presentazione finale dei dati è affidata al sito web *www.floweather.com*.

In esso sono presenti cinque pagine:

1. *Home*: contiene una descrizione generale del dispositivo e delle sue principali caratteristiche;
2. *Blog*: raccoglie una serie di articoli inerenti al progetto;
3. *Project*: descrive in maniera dettagliata componenti e metodi impiegati, consente inoltre di scaricare il codice prodotto;
4. *MyFloweather*: permette l'accesso al servizio personalizzato di presentazione delle misure raccolte;
5. *Credits*: offre una breve panoramica degli sviluppatori.

Attraverso la pagina *MyFloweather* ciascun utente è in grado, dopo aver effettuato una semplice registrazione, di accedere alla sezione personale contenente l'interfaccia di visualizzazione dei dati.

All'interno della pagina sono presenti tre sezioni: 'account', 'weather measurements' e 'air quality measurements'.

Nella prima sono raccolte le informazioni basilari di ogni stazione; essa si articola in due fondamentali schermate chiamate 'station' e 'location'. In quella principale sono inseriti i dati più recenti, affinché possano essere consultabili in maniera immediata: accanto ai valori relativi all'ultima misurazione disponibile, è presente una tabella che riassume alcuni indici (dew point, summer index, heat index), ricavabili a partire dai dati a disposizione, con lo scopo di fornire un'indicazione sul comfort ambientale. La schermata 'location' provvede invece ad indicare su una mappa la posizione della stazione.

La seconda e la terza sezione si occupano invece di presentare graficamente i dati, raggruppati rispettivamente in misurazioni climatiche e di qualità dell'aria. Ciascuna delle quattro variabili calcolate è rappresentata attraverso due schermate: una, oraria, che raccoglie le singole misurazioni mostrandone l'evoluzione nel tempo; l'altra, giornaliera, che fornisce i valori massimi minimi e medi stimati quotidianamente nell'arco delle ventiquattro ore e riassume i valori di riferimento del mese corrente.

Il primo grafico riassume quindi la totalità dei dati a disposizione, a partire dalla messa in funzione del dispositivo, presentandoli con la granularità più fine e in modo da poter apprezzare le variazioni che avvengono su base oraria.

Il secondo grafico, al contrario, effettua una strutturazione più complessa calcolando, a partire dalle ventiquattro misurazioni svolte ogni giorno, gli estremi positivi e negativi oltre a stimarne

la media. Anche in questo caso viene mostrata l'evoluzione temporale delle suddette misure, così da evidenziarne le variazioni. Accanto a questo grafico ne è posto un altro, che riassume gli stessi indici ma con riferimento mensile.

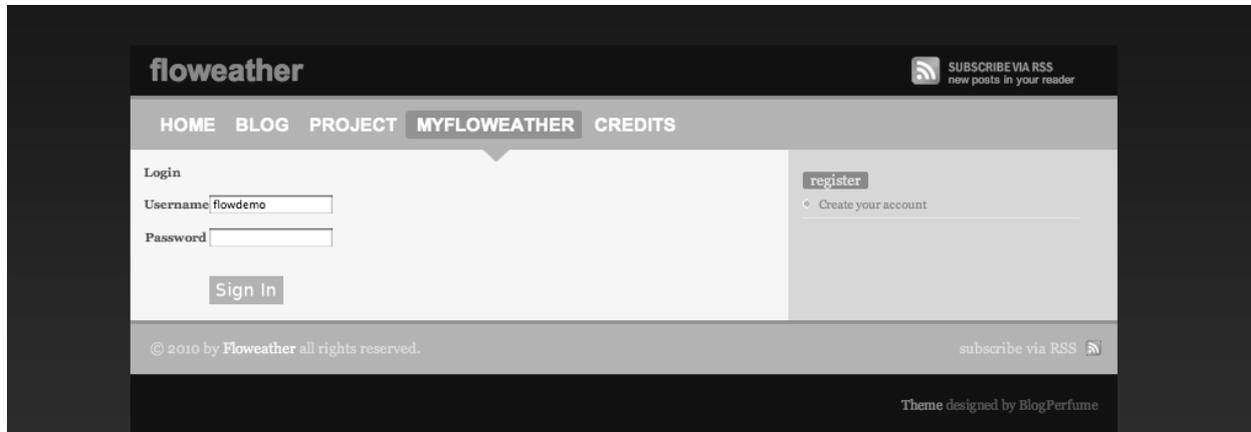


Figura 3.13: Schermata web: floweather.com/myfloweather/login

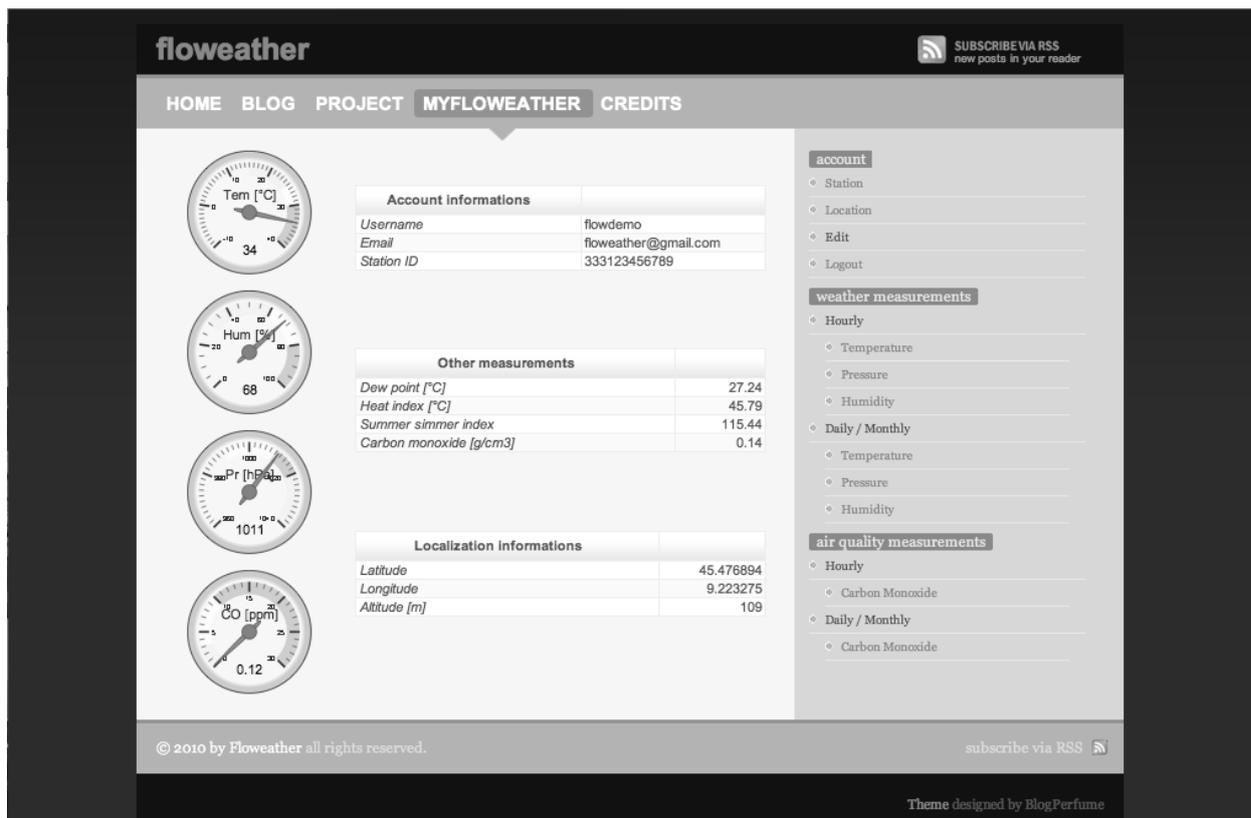


Figura 3.14: Schermata web: floweather.com/myfloweather/station

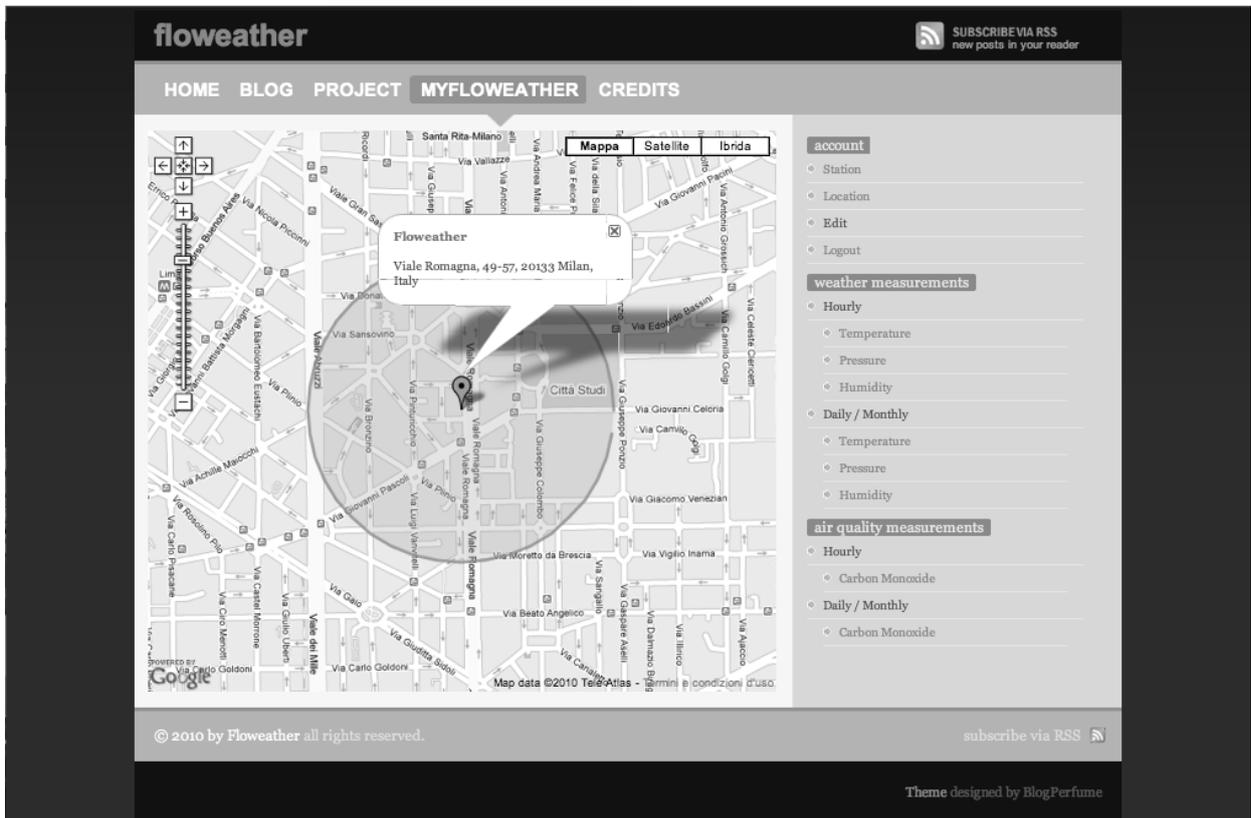


Figura 3.15: Schermata web: floweather.com/myfloweather/location

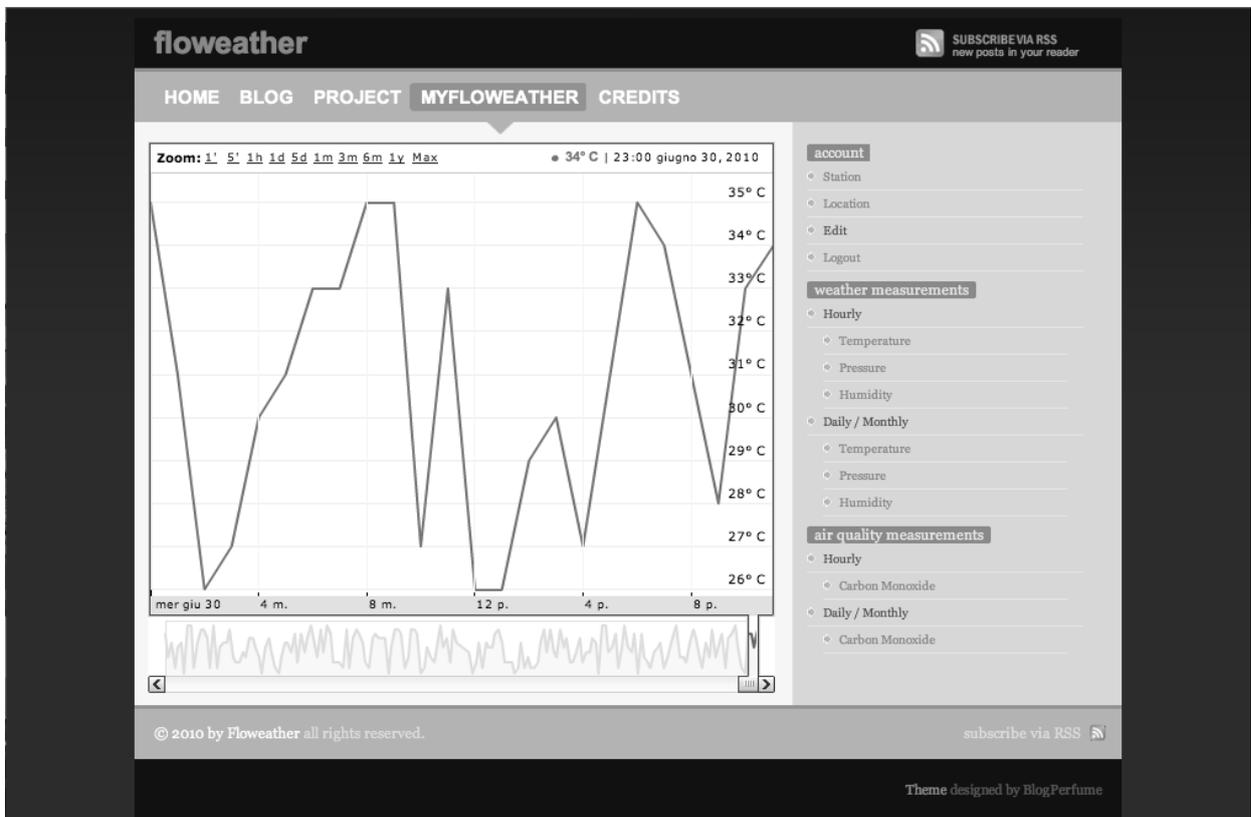


Figura 3.16: Schermata web: floweather.com/myfloweather/hTem

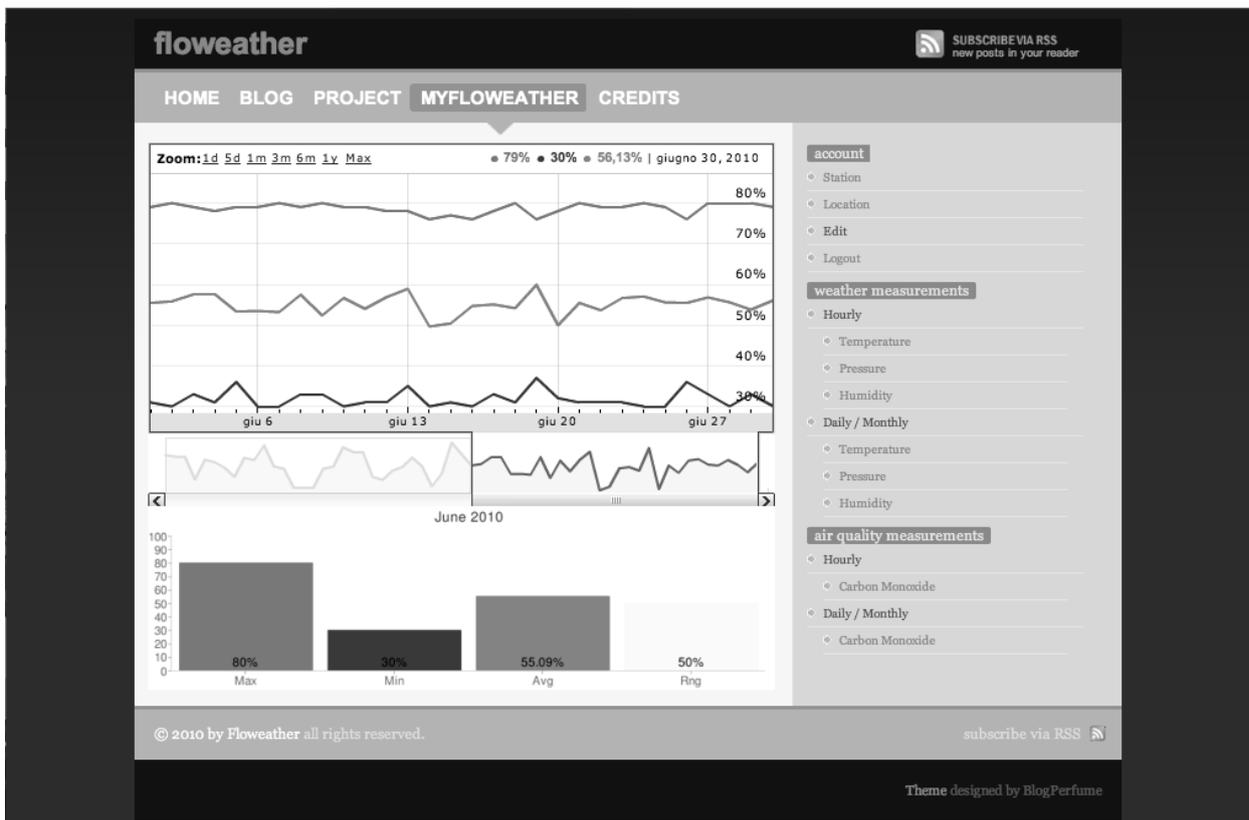


Figura 3.17: Schermata web: floweather.com/myfloweather/dmTem



Figura 3.18: Schermata web: twitter.com/floweather

### **3.3.2.3 Utilizzo all'interno del social network Twitter**

Un'originale funzionalità è quella che permette di condividere le proprie misurazioni connettendo floweather al proprio account Twitter, il famoso social network che permette di condividere informazioni tra persone che hanno gli stessi interessi.

Twitter [21] è un servizio gratuito di microblogging che fornisce agli utenti una pagina personale aggiornabile tramite messaggi di testo con una lunghezza massima di centoquaranta caratteri. Gli aggiornamenti possono essere effettuati tramite il sito stesso, via SMS, e-mail oppure tramite varie applicazioni basate sulle API di Twitter. Gli aggiornamenti sono mostrati istantaneamente nella pagina di profilo dell'utente e comunicati agli utenti che si sono registrati per riceverli.

La filosofia di FLOWEATHER si sposa bene con questo servizio, dal momento che consentirebbe a chiunque di creare un piccolo servizio meteorologico aperto a tutti i vicini e costantemente aggiornato. I post vengono generati automaticamente ogni qual volta giunge una nuova misurazione al server, quindi non si rende necessario l'intervento diretto dell'utente, il quale dovrà unicamente provvedere a collegare i due account la prima volta.

### **3.3.2.4 Utilizzo all'interno di software di simulazione energetica**

L'ultima interessante funzionalità consiste nella possibilità di utilizzare i dati climatici raccolti nel tempo per realizzare simulazioni energetiche grazie all'impiego di software per la progettazione urbanistica.

L'insieme di tutte le misurazioni è infatti reso disponibile in formato EPW, ed è scaricabile gratuitamente dalla propria pagina.

Tale formato, sviluppato specificatamente per dati climatici, è stato creato come standard per i due maggiori programmi di simulazione: ESP ed EnergyPlus. Per le unità di misura è stato scelto il Sistema Internazionale. Il formato appare semplice, del tutto simile a un documento di testo, ed utilizza la virgola come carattere separatore dei dati per facilitarne la lettura.

A partire da questo file è possibile svolgere ogni tipo di simulazione mirata ad una progettazione edilizia che tenga conto degli aspetti energetici, con la differenza che al posto di doversi relazionare con i valori climatici raccolti dalle stazioni poste negli aeroporti, quindi spesso notevolmente distanti dai centri urbani, i dati in questione sono strettamente attinenti al territorio nel quale devono essere impiegati. Così facendo si riescono ad ottenere risultati molto più precisi, migliorando di conseguenza l'efficienza energetica raggiungibile.

# Conclusioni

Il lavoro presentato è consistito nell'implementazione di un originale ed innovativo servizio per il monitoraggio ambientale, che è stata affrontata da diversi punti di vista. L'idea di base era di realizzare un prototipo che fosse in grado di rivoluzionare il concetto di raccolta e presentazione di dati climatici e d'inquinamento, studiato nell'ottica di una possibile commercializzazione futura.

Per questo motivo, le problematiche con cui ci si è dovuti confrontare hanno riguardato non soltanto gli aspetti di progettazione hardware/software, tipici dei sistemi embedded e delle reti wireless di sensori, ma anche quei risvolti propri della produzione industriale come l'impatto sul costo finale ed il design di progetto.

Avendo scelto di lavorare con la piattaforma open source Arduino, ci si è dovuti confrontare inoltre con le caratteristiche specifiche dell'hardware, come la memoria estremamente esigua (2kB) e l'impossibilità di eseguire programmazione *multi tread*.

Per quanto concerne l'infrastruttura informatica sviluppata per la gestione dei dati di ogni stazione, si è cercato di scrivere la maggior parte del codice sotto forma di script *PHP*, che viene eseguito integralmente lato server, limitando all'utilizzo di *Javascript* (che al contrario viene caricato dalla macchina) solo quelle *Applet* per la visualizzazione dei dati, scritte appositamente con tale linguaggio. In questo modo si è riuscito a rendere più agile, e di conseguenza più veloce, il caricamento della pagina web.

Gli ambiti più legati ad una eventuale commercializzazione sono stati presi in considerazione, anche se affrontati solo parzialmente. Trattandosi infatti di una implementazione prototipale, tale sviluppo ha rappresentato la prima tra le numerose fasi che dovranno essere in seguito realizzate per poter arrivare ad un prodotto finito disponibile per il mercato.

L'obiettivo finale che ci si era prefisso, quello di creare un prototipo stabile, funzionale e testato sul medio periodo è stato raggiunto con successo; FLOWEATHER infatti ha mostrato un comportamento affidabile e robusto nel corso dei numerosi test cui è stato sottoposto, fornendo misurazioni sempre coerenti e senza manifestare malfunzionamenti di sorta.

I risultati ottenuti in merito alla sperimentazione di un singolo dispositivo, naturalmente, andrebbero estesi al caso di un maggior numero di stazioni disposte in punti strategici all'interno di una data area urbana che si vuole esaminare, effettuando un'analisi più completa e creando una mappa del clima e dell'inquinamento cittadino.

Le maggiori potenzialità, che sono state solamente intraviste, riguardano, infatti, la possibilità di creare una rete distribuita composta da numerosi dispositivi, così da mettere in relazione i dati raccolti da ciascuna stazione e aggregarli in un'unica struttura di visualizzazione.

Per ciò che concerne gli sviluppi futuri del progetto, una prima direttrice potrebbe essere rappresentata dalla realizzazione dei moduli aggiuntivi precedentemente individuati, la cui integrazione provvederebbe a completare le funzionalità del sistema, oltre ad aprire nuove prospettive di utilizzo.

Un altro degli aspetti sui quali sarà necessario intervenire in futuro è quello concernente il collegamento elettrico tra i vari componenti elettronici. Alla breadboard assemblata per l'occasione, tipica della realizzazione prototipale, verrà sostituito un circuito stampato o PCB, il quale, oltre a garantire una qualità indubbiamente superiore, produrrebbe una sostanziale riduzione degli ingombri. Anche dal punto di vista dei costi, questa scelta comporterebbe un vantaggio, riducendo il prezzo finale, specie nell'ottica di una produzione in scala.

La creazione di un microcontrollore che si possa sostituire alla scheda Arduino costituisce un'altra affascinante idea, e consentirebbe di progettare un componente studiato ad hoc per venire incontro alle particolari esigenze relative alla tecnologia impiegata. Così facendo, si disporrebbe di un hardware ottimizzato e particolarmente adatto per quelle applicazioni mobili che devono mantenere sotto controllo i propri consumi.

La possibilità di mettere un commercio un prodotto consumer come FLOWEATHER appare realistica, data l'assenza sul mercato di un sistema con simili caratteristiche e funzionalità, e considerato l'interesse sempre crescente che si sta sviluppando attorno alle tematiche riguardanti l'ambiente e la qualità della vita dei cittadini nei centri urbani.

I campi di applicabilità di una piattaforma di questo genere potrebbero essere molteplici, spaziando dalla coltivazione agricola al monitoraggio climatico effettuato in ambienti critici, dalla rilevazione di incendi al controllo sui cambiamenti climatici che investono gli habitat naturali al fine di proteggere le specie a rischio di estinzione (sia animali che vegetali). Basti pensare ai vantaggi che si potrebbero ottenere nel ramo dell'enologia, dove una precisa analisi di fattori quali temperatura, umidità e pressione atmosferica rappresenta un elemento essenziale tanto nella fase di viticoltura quanto in quella di conservazione in cantina.

Come sottolineato nei capitoli precedenti, pur non rappresentando l'obiettivo primario del progetto, uno dei possibili utilizzi è quello concernente la raccolta e l'analisi scientifica dei dati. Infatti, per quanto il dispositivo in questione si discosti dalla classica definizione di nodo di una rete di sensori (in quanto più complesso, voluminoso e indubbiamente costoso), esso tuttavia possiede di fatto le caratteristiche di autolocalizzazione e di controllo dei consumi, oltre a realizzare il rilevamento e la trasmissione dei dati seguendo i principi dei comuni sensori.

Da questo punto di vista, un'ipotesi interessante è quella che prevede l'impiego delle stazioni come semplici nodi di una rete, abilitata dallo sfruttamento di un protocollo di comunicazione a corto raggio per trasmissioni in radiofrequenza (come ad esempio ZigBee), da sostituire al modulo GSM. In questo modo, i diversi dispositivi disseminati nell'ambiente sarebbero in grado

di comunicare direttamente tra loro, così da consentire un utilizzo tipico delle reti wireless di sensori.

In conclusione, si può affermare che gli sforzi volti alla progettazione e all'implementazione del prototipo FLOWEATHER hanno prodotto i risultati che erano stati prefissati preliminarmente, presentando un dispositivo che si pone in maniera innovativa nei confronti delle misurazioni ambientali, ed in particolare dei dati climatici.

Le quattro prospettive principali che hanno guidato lo sviluppo del progetto, ossia l'interesse per l'ambiente, i costi ridotti, la filosofia open source e la sostenibilità, provvedono a fornire una caratterizzazione esclusiva e originale, conferendo in questo modo un valore aggiunto al lavoro.

La presenza di un'infrastruttura informatica, creata per il servizio di gestione e visualizzazione dei dati, completa il sistema offrendo l'opportunità di presentare i dati agli utenti in maniera del tutto personale, permettendone la consultazione in tempo reale.

Le prospettive future possono essere innumerevoli, per lo più mirate ad estendere le funzionalità della stazione e a compiere quei passi nella prospettiva di un percorso che porti il prototipo a diventare un prodotto.

# Appendice A

## A.1. Codice Real Time Clock BV4236

```
/*
*****
***** getTime *****
*****/
void BV4236::getTime(int*year, int*month,int*date,int*day,int*hour,int*min) {
    char temp [7];
    int i = 0;
    resetHT(); // reset HT bit
    Wire.beginTransmission(ADDRESS);
    Wire.send(0x2); // start from 02h register
    Wire.endTransmission();
    Wire.requestFrom(ADDRESS, 6); // request 6 next registers
    while(Wire.available()) // check for data from slave
    {
        temp [i] = Wire.receive(); // receive a byte as character
        i++;
    }
    *min = temp [0];
    *hour = temp [1];
    *day = temp [2]; // day of week
    *date = temp [3];
    *month = temp [4];
    *year = temp [5];
    return;
}
```

## A.2. Codice Sensore MPX4115

```
/*
*****
***** MPX4115 *****
*****/
MPX4115::MPX4115(int dataPin) {
    this->dataPin = dataPin;
}

/*
*****
***** getPrs *****
*****/
float MPX4115::getPrs() {
    int adcValue = analogRead(dataPin); // read the output value of MPX4115 from dataPin
    float vout = adcValue * 5.00 /1023; // convert adcValue [0-1023] in Volts
    float prs = ((vout / 5.0) + 0.095) / 0.0009; // compute absolute pressure in mbar or hPa
    return prs;
}
```

### A.3. Codice Sensore SHT15

```
/*
*****
***** SHT15 *****
*****/
SHT15::SHT15(int dataPin, int clockPin) {
    this->dataPin = dataPin;
    this->clockPin = clockPin;
    tem = 0; // Store temperature to compute corrected relative humidity
             // avoiding another temperature computation in getHum()
}

/*
*****
***** getTem *****
*****/
float SHT15::getTem() {
    int temRaw; // Raw value returned from sensor
    // Conversion coefficients from SHT15 datasheet
    const float D1 = -40; // for 14 Bit @ 5V
    const float D2 = 0.01; // for 14 Bit DEGC
    if (sendCommand(CMDTEM, dataPin, clockPin)) // No faults during transmission
    {
        if (getData(dataPin, clockPin, &temRaw)) // Get raw value from sensor
        {
            tem = (temRaw * D2) + D1; // Convert raw value to Celsius
        }
        else tem = -100; // Default value in case of faults
    }
    else tem = -100; // Default value in case of faults
    return (tem);
}

/*
*****
***** getHum *****
*****/
float SHT15::getHum() {
    int humRaw; // Raw humidity value returned from sensor
    float humLin; // Humidity with linear correction applied
    float hum; // Temperature-corrected humidity
    // Conversion coefficients from SHT15 datasheet
    const float C1 = -4.0; // for 12 Bit
    const float C2 = 0.0405; // for 12 Bit
    const float C3 = -0.0000028; // for 12 Bit
    const float T1 = 0.01; // for 14 Bit @ 5V
    const float T2 = 0.00008; // for 14 Bit @ 5V
    if (sendCommand(CMDHUM, dataPin, clockPin)) // No faults during transmission
    {
        if (getData(dataPin, clockPin, &humRaw)) // Get raw value returned from sensor
        {
            humLin = C1 + C2 * humRaw + C3 * humRaw * humRaw; // Apply linear conversion to raw value
            if (tem==0) tem = getTem(); // If there aren't measurement get current temp
            hum = tem * (T1 + T2 * humRaw) + humLin; // Correct humidity value for current temperature
        }
        else hum = -100;
    }
    else hum=-100; // Default humidity value in case of faults
    return (hum); // Default humidity value in case of faults
}
```

## A.4. Codice Sensore TGS2442

```
/*
*****
***** TGS2442 *****
*****/
TGS2442::TGS2442(int heaterPin, int circuitPin, int sensingPin) {
    hPin=heaterPin;
    cPin=circuitPin;
    sPin=sensingPin;
    pinMode(hPin,OUTPUT);
    pinMode(cPin,OUTPUT);
}

/*
*****
***** getCox *****
*****/
float TGS2442::getCox(float tem, int nSamples) {
    this->nSamples=nSamples;
    float V1 = heatAndSense();
    float R1 = 10000.0; //load resistance 10kOhm
    float R0 = ((5.0 * R1) /2.5) - R1; //sensing resistance at 100 ppm [Ohm]
    float Rs = ((5.0 * R1) / V1) - R1; //sensing resistance at actual ppm [Ohm]
    float Rs1 = Rs / getKTem(tem); //Rs temperature compensated [Ohm]
    float RRatio = Rs1/R0; //resistance ratio RS/R0
    float cox = 124/(RRatio - 0.6);
    return cox;
}

/*
*****
***** heatAndSense *****
*****/
float TGS2442::heatAndSense() {
    int count=0;
    int adcValue=0;
    while(count<nSamples)
    {
        // Heating phase (14 ms)
        digitalWrite(hPin,HIGH);
        delay(14);
        // No heat phase (986ms)
        digitalWrite(hPin,LOW);
        delay(981);
        // Sensing phase
        digitalWrite(cPin,HIGH);
        delay(2);
        if (count==nSamples-1) {adcValue = analogRead(sPin);} //last cycle
        delay(3);
        digitalWrite(cPin,LOW);
        count++;
    }
    float V1 = adcValue * 5.00 /1023; //voltage across load resistance
    return V1;
}
```

## A.5. Codice Modulo GM862

```
/*
*****
***** GM862 *****
*****
*/
GM862::GM862(NewSoftSerial *modemPort, int onOffPin) {
    this->modemPort = modemPort;
    modemPort->begin(BAUD_RATE);
    this->onOffPin = onOffPin;
    pinMode(onOffPin, OUTPUT);
}

/*
*****
***** switchOn *****
*****
*/
void GM862::switchOn() {
    digitalWrite(onOffPin, HIGH);
    myDelay(1100);
    digitalWrite(onOffPin, LOW);
    myDelay(2000);
}

/*
*****
***** switchOff *****
*****
*/
void GM862::switchOff() {
    sendAndReceive("AT#SHDN");
}

/*
*****
***** init *****
*****
*/
void GM862::init(char* PIN) {
    //PIN
    strcpy(this->PIN, PIN);
    strcpy(cmd, "AT+CPIN=");
    strcat(cmd, PIN);
    sendAndReceive(cmd);
    //BAUD RATE
    strcpy(cmd, "AT+IPR=");
    strcat(cmd, BAUD_RATE_STRING);
    sendAndReceive(cmd);
}

/*
*****
***** isRegistered *****
*****
*/
boolean GM862::isRegistered() {
    char regStatus;
    sendAndReceive("AT+CREG?"); // is registered?
    regStatus = buf[11];
    if (regStatus == '1')
        return (true);
    else return (false);
}
```

```

/*****
***** sendSMS *****
*****/
void GM862::sendSMS(char *recipient, char *message) {
    strcpy(cmd, "AT+CMGS=");
    strcat(cmd, "\\");
    strcat(cmd, recipient);
    strcat(cmd, "\\");
    sendAndReceive(cmd);
    modemPort->print(message);
    modemPort->print(0x1a, BYTE);    //ctrl-Z char termination
    get(buf);
}

/*****
***** getDataCell *****
*****/
void GM862::getDataCell(char *lac, char *cid) {
    sendAndReceive("AT#MONI");
    int i=0;
    int j=0;
    while(i<strlen(buf))
    {
        if (buf[i]=='L' && buf[i+1]=='A' && buf[i+2]=='C' && buf[i+3]==':')
        {
            i=i+4; break;
        }
        //find LAC: pattern
        i++;
    }
    j=i+4;
    while(i<j)
    {
        *lac+=buf[i];
        i++;
    }
    *lac=0;
    *cid=0;
    while(i<strlen(buf))
    {
        if (buf[i]=='I' && buf[i+1]=='d' && buf[i+2]==':')
        {
            i=i+3;
            break;
        }
        //find Id: pattern
        i++;
    }
    j=i+4;
    while(i<j)
    {
        *cid+=buf[i];
        i++;
    }
    *cid=0;
    return;
}

```

```

/*****
***** initGPRS *****/
/*****/
void GM862::initGPRS() {
    sendAndReceive("AT+CGDCONT=1,\"IP\", \"internet.wind\", \"0.0.0.0\",0,0");
    sendAndReceive("AT#USERID=\"");
    sendAndReceive("AT#PASSW=\"");
}

/*****
***** GPRISActivated *****/
/*****/
boolean GM862::GPRISActivated() {
    sendAndReceive("AT#GPRS?");
    if (strstr(buf,"1")) return true;
    else return false;
}

/*****
***** EnableGPRS *****/
/*****/
void GM862::enableGPRS() {
    sendAndReceive("AT#GPRS=1");
}

/*****
***** OpenSocket *****/
/*****/
boolean GM862::openSocket() {
    sendAndReceive("AT#SKTD=0,80, \"www.flowweather.com\"\\r");
    if (strstr(buf,"CONNECT")) return true;
    else return false;
}

/*****
***** WriteAndReadSocket *****/
/*****/
boolean GM862::writeAndReadSocket(char *data) {
    strcpy(cmd, "GET_/archive/writeXML.php?");
    strcat(cmd, data);
    strcat(cmd, "_HTTP/1.1\\r\\n");
    strcat(cmd, "HOST:_www.flowweather.com\\r\\n\\r\\n");
    sendAndReceive(cmd);
    if (strstr(buf,"200")) return true;
    else return false;
}

/*****
***** MyDelay *****/
/*****/
void GM862::myDelay(int timeout) {
    long previousMillis=millis();
    while (millis() - previousMillis < timeout) {}
    return;
}

```

# Bibliografia

- [1] [http://it.wikipedia.org/wiki/Wireless\\_sensor\\_network](http://it.wikipedia.org/wiki/Wireless_sensor_network)
- [2] Sensor Networks - 2009 A. Capone - Dip. di Elettronica e Informazione - Politecnico di Milano
- [3] <http://alert.dot.pima.gov>
- [4] [http://en.wikipedia.org/wiki/Embedded\\_system](http://en.wikipedia.org/wiki/Embedded_system)
- [5] Introduzione ai sistemi embedded - 2009 A. Bardine, C.A. Prete - Dip. Ingegneria Informazione - Università di Pisa
- [6] Introduzione ai sistemi di misura distribuiti - Facoltà di ingegneria - Università degli studi di Cassino
- [7] <http://apps1.eere.energy.gov/buildings/energyplus/>
- [8] <http://www.eere.energy.gov/>
- [9] [http://apps1.eere.energy.gov/buildings/tools\\_directory/software.cfm/ID=123/](http://apps1.eere.energy.gov/buildings/tools_directory/software.cfm/ID=123/)
- [10] EnergyPlus - 2009 US Department of Energy - Guide for interface developers
- [11] Il polietilene - Corso di tecnologia dei polimeri - 2005 S. Esposito - Università degli studi di Cassino
- [12] <http://creativecommons.org/licenses/by-sa/2.5/>
- [13] <http://arduino.cc/>
- [14] [http://www.sensirion.com/en/pdf/product\\_information/Datasheet-humidity-sensor-SHT1x.pdf](http://www.sensirion.com/en/pdf/product_information/Datasheet-humidity-sensor-SHT1x.pdf)
- [15] [http://www.freescale.com/files/sensors/doc/data\\_sheet/MPX4115.pdf](http://www.freescale.com/files/sensors/doc/data_sheet/MPX4115.pdf)
- [16] <http://www.figarosensor.com/products/2442pdf.pdf>
- [17] [http://www.telit.co.it/en/products/gsm-gprs.php?p\\_id=12&p\\_ac=show&p=4](http://www.telit.co.it/en/products/gsm-gprs.php?p_id=12&p_ac=show&p=4)
- [18] <http://www.i2c.byvac.com/downloads/BV4236%20DataSheet.pdf>

[19] <http://www.cadsoftusa.com/index.htm.en>

[20] <http://www.stack.nl/~dimitri/doxygen/>

[21] <http://twitter.com/>

# Ringraziamenti

Non avrei mai pensato di dover ringraziare tutte queste persone per un vaso, ma il loro apporto è stato indispensabile e mi ha dato lo slancio per arrivare in fondo.

Ho conosciuto Stefano Manni quasi per caso, ma ho sempre ringraziato la provvidenza per quell'incontro; senza di lui non sarei stato in grado di realizzare ciò che avevo in mente né di farlo così bene. Ha sottratto tempo ai suoi studi e al tempo libero per questo progetto dimostrandosi una persona responsabile e leale, con cui sono onorato di avere lavorato.

Quando per la prima volta ho proposto la mia idea di tesi al Professore Capone ho ricevuto da subito un enorme supporto, oltre all'avviso 'Non hai idea del casino in cui ti stai cacciando'. Diciamo che un'idea me la sono fatta col tempo, ma la fiducia concessami, oltre al supporto economico e ai numerosi consigli, sono stati fondamentali e mi hanno consentito di apprezzare l'uomo prima che il docente.

Su Toti è difficile dire qualcosa di nuovo visto che da quindici anni cerchiamo disperatamente di continuare a fare tutto insieme, sebbene se le nostre strade col tempo non siano rimaste parallele; anche questa volta ha avuto un ruolo fondamentale, il suo talento ha permesso di rendere il progetto più bello di quanto realmente fosse.

Definire il mio stile di scrittura contorto e macchinoso è un eufemismo, eppure Claudia non ha fatto una piega correggendo a distanza oltre un centinaio di pagine, neanche si trattasse della sua tesi. Le ho sottratto davvero un sacco di tempo e non le sono stato vicino come meritava in questo periodo, prometto che da adesso comincerò a farmi perdonare.

Se dopo tanti sforzi sono diventato un ingegnere lo devo soprattutto a mio padre, che mi ha insegnato, tra mille cose, ad essere onesto e responsabile. La sua figura mi ha guidato sin da piccolo, ed è per merito suo che avevo le idee chiare sul mio futuro sin dalla quarta elementare.

Se sono sopravvissuto lontano da casa in questi anni lo devo soprattutto a mia madre, da cui ho imparato ad essere indipendente, determinato e ad inseguire i miei sogni. Le telefono sempre meno di quanto dovrei, ma è l'unica mamma che può dire 'E non studiare sempre, ogni tanto esci e divertiti'.

L'artista di famiglia è sempre stata mia sorella e, anche se le sue abilità sono di ben altro livello, mi è stata di enorme aiuto nel sistemare le immagini o, in alcuni casi, a disegnarle di nuovo.

Mio fratello non mi ha aiutati direttamente, ma è finalmente riuscito a diplomarsi, e per questo merita una citazione.

Un anno in CDS non si scorda tanto facilmente. Gli amici con cui sono stato fianco a fianco ogni giorno per tutto questo tempo sono stati come una famiglia, e mi hanno permesso quando dovevo massacrarmi di studio di farlo sempre col sorriso sulle labbra. Non li elencherò tutti perchè sono davvero troppi, ma ciascuno di loro ha contribuito a farmi passare uno dei periodi più belli della mia vita.

Ho lasciato molti amici a Palermo, altri ne ho incontrati qui, altri ancora sono sparpagliati in giro: tutti mi hanno sempre dato una mano, semplicemente standomi a sentire, o addirittura venendo a Milano solo per assistere ad una discussione su un vaso da fiori, e per questo mi considero realmente fortunato.

La mia rincorsa alla laurea di quest'anno contro la stagione quasi perfetta di Fabrizio Miccoli è un paragone che non regge. Ma è l'unico paradigma per raccontare il filo conduttore che accompagna le mie vicende a quelle del Palermo, il solo modo che possa spiegare cosa è per me il rosanero.

Ultimo ringraziamento a me stesso: hai avuto coraggio e testardaggine ed hai concluso anche questa, pur prendendotela con comodo; fin qui te le sei spassata, adesso è ora di decidere cosa fare da grande, da domani sono cazzi tuoi.