



UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA

Tesi di Laurea Triennale in Informatica (L-31)

Monitoraggio ambientale in ambito smart city

Relatore

Prof. Esposito Christiancarmine

Candidato

Gerardo Sessa

Matr. 05125 07759

ANNO ACCADEMICO 2021-2022

Abstract

Nel mondo sempre più connesso di oggi, l'Internet of Things (IoT) sta giocando un ruolo fondamentale. I contesti di applicazione sono i più svariati; uno di fondamentale interesse per i cittadini è quello delle *Smart City*. Una Smart City (città smart) è «intelligente» nell'utilizzo delle informazioni recepite dagli stessi cittadini sfruttando i loro dispositivi interconnessi a tecnologie IoT (Internet of Things), monitorando le zone urbane per motivi economici e di sostenibilità (*es.* risparmio energetico) o di mobilità cittadina (*es.* controllo congestione stradale). Inoltre, il concetto di smart cities è adoperato anche in ambito di sicurezza, educazione, mezzi di sussistenza e ambiente.

Le soluzioni presenti in mercato, in grande maggioranza private, sono caratterizzate da architetture chiuse, che non offrono possibilità di accesso o di ampliamento. L'intento di questo elaborato è l'analisi di un'architettura esistente, come quella proposta da *Libelium*, al fine di capire quali possibili estensioni possono essere sviluppate, favorendo così il concetto di Open Data. L'obiettivo di questo progetto è la realizzazione di un sistema di monitoraggio in ambito Smart City per il rumore, l'umidità, la pressione e la temperatura. Per lo sviluppo del progetto viene adoperata una smart board proprietaria, equipaggiata con appositi sensori, collegata ad un gateway capace di raccogliere le informazioni. Le informazioni saranno poi

trasmesse ad un database esterno per la rielaborazione e la presentazione agli utenti su una piattaforma web.

Questo lavoro è un punto di partenza per lo sviluppo di applicazioni nel contesto dell'IoT capaci di semplificare la vita dei cittadini.

Table of Contents

| | | |
|----------|--------------------------------------------------------------------------|-----------|
| 1 | Smart City - La città intelligente | 7 |
| 1.1 | Da città tradizionale a smart city | 7 |
| 1.2 | Strumenti principali dell'IoT | 8 |
| 1.3 | Cos'è una smart city | 9 |
| 1.4 | Raccolta dei dati | 10 |
| 1.5 | Tipologie di applicazioni di Internet of Things in zone urbane | 12 |
| 1.6 | Esempi di applicazioni di Internet of Things in zone urbane | 14 |
| 1.6.1 | Controllo dell'inquinamento cittadino | 14 |
| 1.6.2 | Gestione di lampioni fotovoltaici | 17 |
| 1.6.3 | Illuminazione stradale sostenibile | 18 |
| 1.7 | Conclusioni | 19 |
| 2 | Protocolli di comunicazione | 20 |
| 2.1 | 802.15.4 | 20 |
| 2.2 | Bluetooth Low Energy - BLE 4.0 | 21 |
| 2.3 | Low Power-Wide Area Network - LP-WAN | 22 |
| 2.3.1 | Sigfox | 22 |
| 2.3.2 | Nb-Iot | 23 |
| 2.3.3 | LoRa | 23 |

| | | |
|----------|-------------------------------------------------|-----------|
| 2.3.4 | Zigbee | 23 |
| 2.4 | Conclusioni | 24 |
| 3 | Monitorare rumore, umidità e temperatura | 25 |
| 3.1 | Hardware | 25 |
| 3.1.1 | Waspote v15 | 26 |
| 3.1.2 | Waspote smart cities PRO | 28 |
| 3.1.3 | Modulo XBee 868LP | 30 |
| 3.1.4 | Meshlium | 31 |
| 3.2 | Software | 34 |
| 3.2.1 | Meshlium Manager System | 34 |
| 3.2.2 | MySQL | 35 |
| 3.3 | Sistema proposto | 37 |
| 3.3.1 | Sensing | 37 |
| 3.3.2 | Trasmissione | 39 |
| 3.3.3 | Salvataggio | 40 |
| 3.3.4 | Visualizzazione | 42 |
| 3.4 | Conclusioni | 43 |
| 4 | Conclusioni | 44 |
| | List of Figures | 45 |
| | Bibliografia | 47 |
| | Dediche e ringraziamenti | 49 |

Introduzione

Per **Internet of Things** (*IoT*) intendiamo l'applicazione di oggetti (le *cose*, da cui il nome) di utilizzo comune in una rete wireless, implementandoli in un ambiente digitale. Un oggetto diventa allora *smart* (intelligente), andando ad interagire con il mondo circostante acquisendo e inviando dati tra rete e mondo reale. E' con queste modalità d'utilizzo che possiamo sfruttare i dati ricavati dagli oggetti per l'utilizzo di strumenti di controllo, automazione e monitoraggio, inoltre ricavare dati ci permette di scegliere con maggiore accuratezza le prossime scelte.

Un ruolo fondamentale nell'IoT è quello ricoperto dai **sensori**, che rilevano i cambiamenti nell'ambiente circostante andando a trasformare in dati digitali le informazioni ottenute, poi condivise ad altri dispositivi collegati alla rete. Possono avvisare di potenziali problemi prima che diventino gravi (*diagnostica predittiva*), consentendo alle aziende di eseguire la *manutenzione predittiva* e quindi, se prendiamo come esempio un contesto aziendale, evitare costosi tempi di fermo-macchine. I dati dei sensori possono anche essere analizzati per consentire di ottenere informazioni su tendenze cruciali e prendere decisioni importanti. I sensori sono disponibili in molte forme e dimensioni, alcuni sono costruiti appositamente e contengono molti sensori individuali incorporati, consentendo di monitorare e misurare molte fonti di dati [1].

Ad oggi, l'applicazione dell'IoT è presente in più ambiti: *domotica*, per gestire la tecnologia applicata alla casa, andando ad interagire ad esempio con elettrodomestici e luci; *auto intelligenti*, che con i loro sensori possono ad esempio parcheggiare in autonomia, *smart city*, per la gestione di dinamiche cittadine, come il funzionamento dei semafori con la conseguente gestione del traffico; *smart health*, come la gestione remota dei pazienti, e così via, tutti messi in funzione tramite interconnessione di oggetti intelligenti.

Ogni persona che vive in una città moderna ha più di un dispositivo in suo possesso con connessione ad internet. Questo significa che un grosso quantitativo di devices e sistemi può potenzialmente essere sfruttato per un'unica grande infrastruttura di rete cittadina. Il concetto alla base delle **smart cities** è di fornire una connessione tra cittadini e servizi. [2]

Una *Smart City* è una zona urbana che utilizza collezioni di dati raccolti da vari tipi di dispositivi e sensori per monitorare e gestire le infrastrutture e risorse cittadine efficientemente. L'utilizzo di dati recepiti permette ai sistemi di adattarsi alle circostanze, riuscendo ad offrire sempre un soddisfacente utilizzo.

L'obiettivo della tesi è quello di sviluppare un sistema di monitoraggio per smart cities con sensori adatti al rilevamento di rumore, umidità, pressione e temperatura. Per raggiungere l'obiettivo è stato necessario dividere il lavoro di tesi in due fasi; nella prima fase, quella di sensing, verrà utilizzato un insieme di sensori, capaci di comunicare con un server esterno attraverso protocolli tipici del contesto dell'IoT, al fine di raccogliere i dati; nella seconda, invece, una piattaforma web per la visualizzazione dei dati raccolti sarà sviluppata. Il documento è strutturato in quattro capitoli: il primo capitolo dà una visione più dettagliata del contesto delle smart cities, con descrizione di alcuni esempi già esistenti; il secondo capitolo tratterà l'aspetto relativo alla comunicazione, andando ad analizzare i differenti protocolli

esistenti nell'ambito dell'IoT, al fine di capire quale è il migliore per il nostro scopo. Nel terzo capitolo verrà presentato il sistema sviluppato, sia nella parte hardware, vedendo le singole componenti nel dettaglio, sia nella parte software. Nel quarto capitolo saranno descritte le conclusioni e gli sviluppi futuri.

1 Smart City - La città intelligente

1.1 Da città tradizionale a smart city

Il concetto di *smart cities* rientra nel tema della **trasformazione digitale**.

La trasformazione digitale è un termine utilizzato per indicare cambiamenti (nel caso preso in esame sociali e urbani) legati all'applicazione di tecnologia digitale. Nel processo di trasformazione digitale vengono coinvolti tutti i partecipanti dell'ecosistema, attraverso la condivisione di informazioni.

L'idea della **smart city** (città intelligente) è quella di gestire le risorse in modo intelligente, mirando a diventare sostenibile economicamente ed energeticamente, al fine di migliorare la qualità di vita dei suoi cittadini.

Grazie all'utilizzo di sistemi tecnologici connessi, è possibile valutare come gestire al meglio le risorse, riducendo le emissioni, sviluppando reti di trasporto urbano sostenibili e andando a studiare soluzioni più efficienti per l'illuminazione e il riscaldamento.

Ad oggi nel mondo sono già state applicate numerose implementazioni urbane con l'IoT.

In Spagna, nella città di *Toledo*, attraverso *Canal Smart*, i cittadini e i turisti hanno modo di segnalare incidenti in città, segnalando il luogo sulla mappa e descrivendo il problema e con possibilità di allegare immagini, in modo da rendere più vivibile il luogo per abitanti e turisti. [3]

Sempre in Spagna, a *Barcellona*, intelligenti bidoni della spazzatura raggiunta un'elevata capienza scaricano i rifiuti nel deposito sotterraneo, riducendo i cattivi odori nelle strade e rilevando quali sono i punti critici della città in cui si accumula più spazzatura.

A *Boston* (Usa), sono state rese disponibili una raccolta di app che contengono informazioni sui parcheggi, strumenti e segnalare problemi di servizio. Inoltre, i cittadini possono rintracciare uno scuolabus nei dintorni o segnalare la presenza di graffiti in qualsiasi punto della città.

A *Oslo* (Norvegia), l'IoT è stato applicato su strada per gestione dell'illuminazione delle luci led in base alle esigenze (tratteremo un esempio simile nel paragrafo dedicato agli esempi di applicazioni), rilevando inoltre la presenza di veicoli su strada per segnalare la presenza di ingorghi ed evitare ulteriore traffico. [4]

Anche in *Italia* si sta iniziando a promuovere strategie per rendere le città del nostro paese smart, con sempre più comuni presenti nelle nostre zone improntati a realizzare piattaforme per servizi pubblici online e avere punti d'accesso al wi-fi pubblico.

1.2 Strumenti principali dell'IoT

L'idea alla base dell'*Internet of things* è quella di interconnettere più dispositivi eterogenei in un'ambiente digitale in modo da poter comunicare tra di loro le informazioni ottenute per elaborare soluzioni ottimali. Prima di procedere, stabiliamo gli elementi fondamentali che andremo ad osservare per l'utilizzo di un sistema di

IoT.

I sensori IoT hanno il compito di raccogliere, elaborare e trasferire dati. Sono elementi di piccole dimensioni, installati per acquisire parametri come temperatura, umidità, rumore, qualità dell'aria, presenza di fumo e così via. Sono formati da circuiti integrati minuscoli, robusti e capaci di sopravvivere in ambienti ostili. I sensori sono in grado di dialogare col mondo esterno, a cui inviano in real time dati ed informazioni necessarie, quindi devono saper sfruttare le comunicazioni wireless e connettersi alla rete. Per questo motivo andremo a formare una *WSN*, rete wireless di sensori interconnessi.

Attraverso le WSN è possibile monitorare svariati indicatori, come la temperatura e l'umidità negli *smart buildings*, aiutando a rendere più vivibile l'ambiente, oppure si può monitorare il traffico in modo da inviare informazioni ai cittadini alla guida come quale sia il miglior percorso al momento, segnalando congestioni e mantenendo la guida sicura.

1.3 Cos'è una smart city

Nonostante non ci sia una definizione standard del termine **smart city**, possiamo descrivere «l'intelligenza» di una città nella capacità di raccogliere dati attraverso risorse condivise, cioè sfruttando i devices distribuiti nella zona urbana, coinvolgendo i cittadini stessi nella raccolta delle informazioni per la costruzione di una **WSN** (*Wireless sensor network*), con lo scopo di migliorare lo stile di vita degli abitanti attraverso l'automatizzazione della gestione di strade e strutture pubbliche, nonché per sfruttare le informazioni recepite con la monitorizzazione della città per prendere con maggiore accuratezza le nostre prossime scelte.

Per quanto riguarda lo sviluppo di WSN in smart cities, possiamo fare riferimento

principalmente a tre tipologie:

1. **Static WSN**, usate per monitorare aree come strutture ospedaliere, una casa o palazzo. Presenta nodi «stazionari» utilizzati per coprire l'intera area interessata.
2. **Mobile WSN**, in cui almeno un nodo è mobile, come veicoli che trasportano sensori o persino cellulari grazie all'integrazione dei sensori inclusi in uno smartphone. ad esempio si può rendere il veicolo di un cittadino un nodo per il calcolo del traffico.
3. **Crowd sensing**, considerato che uno smartphone può ricevere e condividere più tipi di dati, si va a coinvolgere direttamente l'utente nella raccolta di dati (GPS, camera ecc.). I sensori degli utenti sono però meno precisi di quelli specializzati e hanno maggior margine d'errore. Inoltre potrebbe causare problemi se si applica in modo da rischiare di intaccare la privacy.

1.4 Raccolta dei dati

I processi di IoT si dividono in quattro fasi:

1. **Acquisire i dati**. Mediante i sensori, i dispositivi catturano i dati dall'ambiente in cui sono installati.
2. **Condividere i dati**. Attraverso la rete, i dispositivi IoT hanno la possibilità di inviare i dati a un sistema cloud pubblico o privato (*dispositivo-sistema-dispositivo*) o a un altro dispositivo (*dispositivo-dispositivo*), oppure li archiviano localmente seguendo le istruzioni per l'elaborazione alla periferia della rete.

3. **Elaborare i dati.** Il software viene programmato per svolgere un compito in base ai dati raccolti.
4. **Agire sulla base dei dati.** I dati accumulati da tutti i dispositivi inclusi in una rete IoT vengono analizzati e trasformati in *insight* (approfondimenti basati sulle analitiche ricavate) su cui poggiare future decisioni sicure.[5]

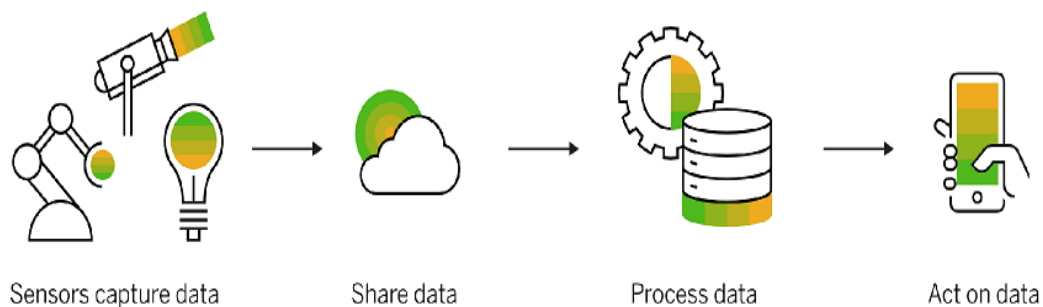


Figura 1.1: Le quattro fasi principali dell'Internet of things

Possiamo monitorare l'integrità delle strutture ricevendo dati sulle vibrazioni di palazzi o ponti. L'estensione dell'utilizzo di dati di monitoraggio in città rende la vita più sicura e comoda.

Le WSN da utilizzare vanno studiate accuratamente, dato che potrebbero insorgere problemi dovuti ad esempio alla limitata capacità dei sensori, considerando quali sensori vogliamo utilizzare, come implementarli per coprire l'area di interesse ottimizzando costi ed efficienza.

A comporre le WSN abbiamo i *sensor nodes*, come può essere **Wasp mote**, che rileva dati con i suoi sensori per poi inviarli ad un dispositivo in rete che si occupa di collezionare i dati. [6]

Per comunicare sulla rete, il sensor node utilizzerà una delle tecnologie come *5G*,

Bluetooth, Wi-Fi, Wi-Fi Pro, XBee 868LP. Un **gateway** ci permette di ricevere dati dai nodi per poi reindirizzarli direttamente su internet, oltre ad essere forniti di un database interno in grado di collezionare i dati. Il gateway rappresenta quindi il punto di accesso dell'utente ad una rete esterna.

1.5 Tipologie di applicazioni di Internet of Things in zone urbane

Prendiamo in esame alcune delle diverse applicazioni dell'utilizzo di nodi e sensori in smart cities:

1. **Structural Heath Monitoring** (*SHM*), utilizzati per rilevare anomalie in una infrastruttura civile e garantire sicurezza, rilevando umidità, temperature e così via. Implementati via l'algoritmo greedy (scelta della soluzione più appetibile per quel determinato programma a ogni passo locale) che sviluppa i nodi step by step, testando anche i worst case. Si studiano due algoritmi greedy per sviluppare i sensor nodes per test modali sulla struttura. Va considerata anche la metrica WSN, inoltre può essere un problema la connettività tra nodi eterogenei, che quindi deve rispettare metrica SHM, WSN lifetime e la connettività. Per SHM esistono due set di nodi: low-end per le risorse obbligatorie e high-end per le risorse aggiuntive (non obbligatorie). E' possibile applicare algoritmi euristici (visto che i tempi di greedy possono essere lunghi) per SHM piccoli, per SHM più ampi usiamo comunque greedy.
2. **Pipeline Network Monitoring**, utilizzati per trasportare acqua, olio, gas. Sono facilmente esposti a danni e contaminazioni, quindi monitorare le pipeline

diventa importante per la protezione ambientale e la sanità pubblica. Possiede metriche tra cui i tempi di rilevazione e le area coperte. Con l'algoritmo greedy calcoliamo il prossimo punto in cui conviene sviluppare un sensore successivo. E' possibile integrare nodi mobili sulla zona di interesse.

3. **Camera sensor Network:** immagini per un'area monitorata. La camera monitora con una visione "a cono", potendosi però muovere in più direzioni. Bisogna allora ottimizzare i costi e le posizioni delle camere per coprire l'area (l'obiettivo è quello di avere al minor costo la maggior copertura). Inoltre si può tracciare con un set di camere che segnano il target e le altre camere lo seguono dinamicamente con un algoritmo greedy.
4. **Urban traffic monitoring,** per ridurre la congestione nelle strade, è importante registrare dati real time del traffico urbano. Generalmente si usano nodi statici, oppure utilizzando gli stessi veicoli come sensor node in grado di connettersi alla WSN della smart city. In questo modo viene reso possibile all'utente di controllare il traffico attraverso il GPS e la velocità della strada. Per preservare la privacy, i veicoli per il calcolo sono per lo più taxi e bus, che servono a costruire una mappa della velocità. Gli smartphone possono essere valutati inoltre come potenziali nodi per l'invio/ricezione di dati GPS.
5. **Streetlight monitoring,** per ridurre i consumi di energia riuscendo a garantire ugualmente sicurezza sulla strada. Ad esempio, possiamo decidere in base a una regione se più o meno abitata quanta luce utilizzare. Inoltre, possiamo dotare i lampioni stradali stessi di sensori per misurare lo stato attuale dell'illuminazione e sfruttare la ricezione di dati real time per la gestione. Maggiori esempi a riguardo nel prossimo capitolo.

1.6 Esempi di applicazioni di Internet of Things in zone urbane

1.6.1 Controllo dell'inquinamento cittadino

L'esempio preso in esame di controllo della qualità dell'aria prevede l'implementazione di un *Waspote Gases Sensor Board* in grado di rilevare *NO*, *CO*, *CO₂* per esaminare l'inquinamento cittadino. Nel caso proposto, utilizzando un modulo *GPRS/GSM Quadband*, il Waspote si riverserà su una stazione radio base (BTS) per inviare i dati rilevati su internet.[7]

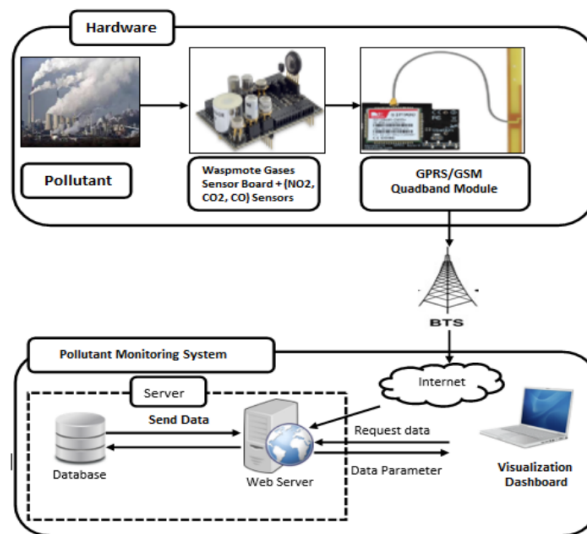


Figura 1.2: Architettura sistema di monitoraggio dell'inquinamento

I valori rilevati, collezionati da un database, vengono presentati all'utente in una web page da cui è possibile visualizzare i grafici dedicati alla temperatura (in celsius) e che notifica il superamento della soglia sicura della presenza di NO, CO, CO2 nell'aria.

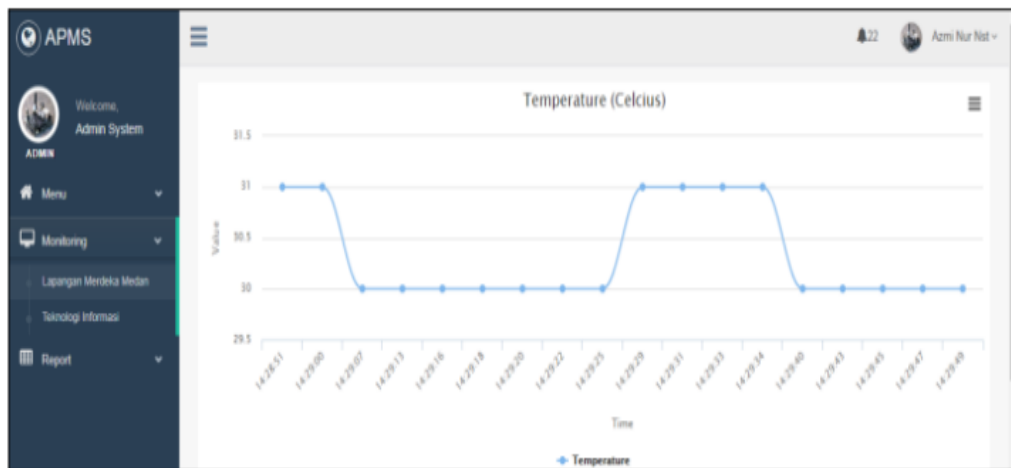
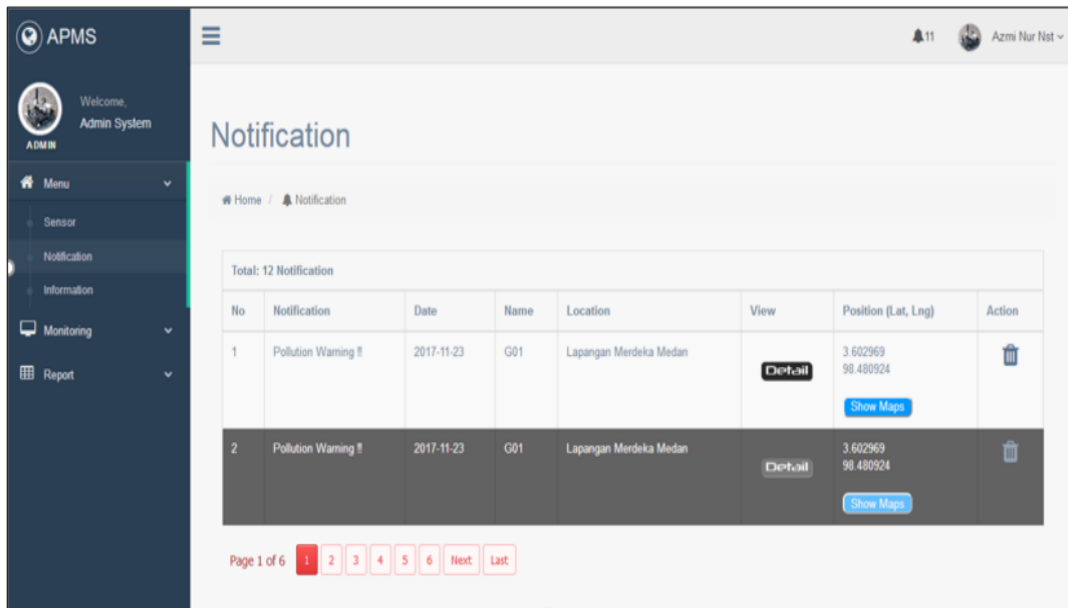


Figura 1.3: Temperature in celsius rilevate nei punti d'interesse

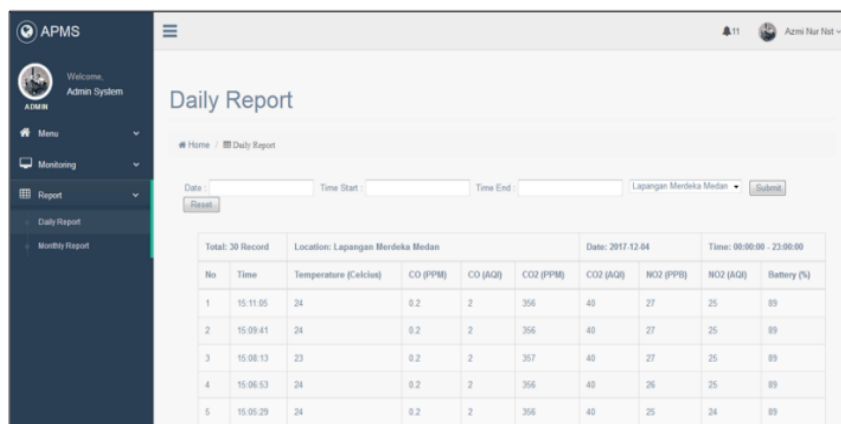


The screenshot shows the 'Notification' page in the APMS Admin System. The left sidebar contains a menu with options: Sensor, Notification (selected), Information, Monitoring, and Report. The main content area displays a table of notifications. The first two rows show 'Pollution Warning #' notifications from 2017-11-23 at 'Lapangan Merdeka Medan' with sensor ID 'G01'. Each row has a 'Detail' button and a 'Show Maps' button. The bottom of the page shows pagination: 'Page 1 of 6' with buttons for 1, 2, 3, 4, 5, 6, Next, and Last.

| No | Notification | Date | Name | Location | View | Position (Lat, Lng) | Action |
|----|---------------------|------------|------|------------------------|--------|-----------------------|-----------|
| 1 | Pollution Warning # | 2017-11-23 | G01 | Lapangan Merdeka Medan | Detail | 3.602969 98.480924 | Show Maps |
| 2 | Pollution Warning # | 2017-11-23 | G01 | Lapangan Merdeka Medan | Detail | 3.602969 98.480924 | Show Maps |

Figura 1.4: Notificazione inquinamento rilevato

Il sistema prevede una pagina per report giornalieri, indicando data, temperatura, i valori dei principali gas da analizzare e il livello della batteria. Oltre ai report giornalieri è possibile visualizzare report mensili per gli stessi valori.



The screenshot shows the 'Daily Report' page in the APMS Admin System. The left sidebar contains a menu with options: Menu, Monitoring, Report (selected), Daily Report (selected), and Monthly Report. The main content area displays a table of daily sensor data. The table has columns for No, Time, Temperature (Celsius), CO (PPM), CO (AQI), CO2 (PPM), CO2 (AQI), NO2 (PPM), NO2 (AQI), and Battery (%). The data is for the date 2017-12-04 at 'Lapangan Merdeka Medan'. The table shows 5 records. The bottom of the page shows pagination: 'Page 1 of 6' with buttons for 1, 2, 3, 4, 5, 6, Next, and Last.

| No | Time | Temperature (Celsius) | CO (PPM) | CO (AQI) | CO2 (PPM) | CO2 (AQI) | NO2 (PPM) | NO2 (AQI) | Battery (%) |
|----|----------|-----------------------|----------|----------|-----------|-----------|-----------|-----------|-------------|
| 1 | 15:11:05 | 24 | 0.2 | 2 | 356 | 40 | 27 | 25 | 89 |
| 2 | 15:09:41 | 24 | 0.2 | 2 | 356 | 40 | 27 | 25 | 89 |
| 3 | 15:08:13 | 23 | 0.2 | 2 | 357 | 40 | 27 | 25 | 89 |
| 4 | 15:06:53 | 24 | 0.2 | 2 | 356 | 40 | 25 | 25 | 89 |
| 5 | 15:05:29 | 24 | 0.2 | 2 | 356 | 40 | 25 | 24 | 89 |

Figura 1.5: Daily Report

1.6.2 Gestione di lampioni fotovoltaici

Riferendoci al monitoraggio della luce stradale citato nel capitolo precedente, possiamo ricevere informazioni sulla batteria di un lampione fotovoltaico andando a collegare i sensori ai componenti per misurare la tensione e il flusso di corrente di pannello solare, batteria e lampione. Tutti questi sensori vengono controllati utilizzando *WASPMOTE V1.1* in modo che i dati grezzi (*raw data*) vengano inviati via comunicazione wireless al PC di monitoraggio che prevede *Protocollo GSM* e *Zigbee*.^[8]

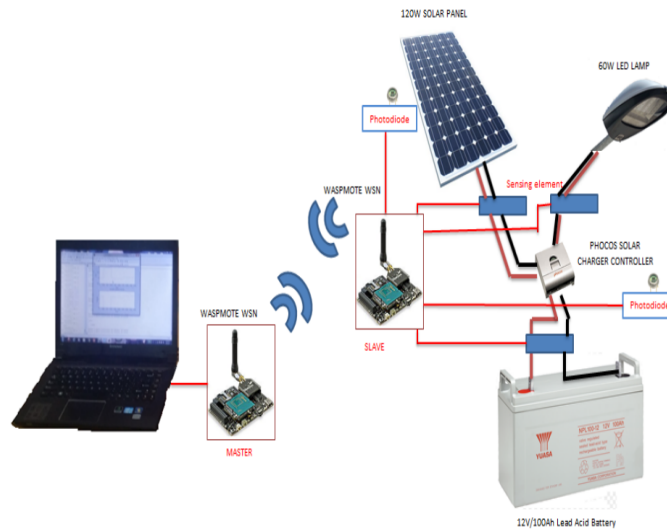


Figura 1.6: Setup per il monitoraggio di un lampione fotovoltaico

I dati sulla batteria vengono inviati attraverso una comunicazione wireless *GSM* e mostrati in real time mediante l'utilizzo del software MATLAB, utilizzato per l'analisi delle informazioni ottenute attraverso grafici. I risultati sono espressi in tempo (indicato dai giorni) e voltaggio della batteria osservata.

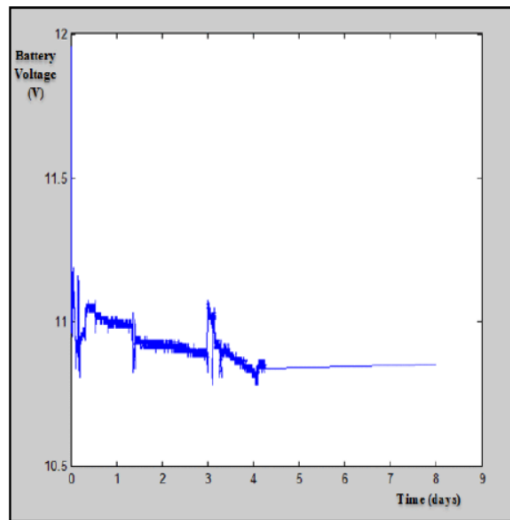


Figura 1.7: Monitoraggio batteria visualizzato da MatLab

1.6.3 Illuminazione stradale sostenibile

Sempre sul tema dell'illuminazione stradale, un'altra interessante realizzazione di gestione cittadina intelligente è quello della Malesia, dove si è pensato, attraverso una WSN, di installare lampioni dotati di waspnote negli orari notturni settati al 40% della luminosità finchè non rivelano movimento, al che tornano a fornire il massimo della luce sulla strada per un determinato periodo, garantendo la viabilità. In questo esempio, i nodi si affacciano ad un gateway, il **meshlium**, per ampliare il range di comunicazione fino al PC che fa da server.[9]

Questo tipo di implementazione urbana è una scelta green per ridurre i consumi, grosso vantaggio a favore delle smart cities.

1.7 Conclusioni

Una volta comprese le meccaniche alla base della raccolta dei dati nel campo del IOT, le applicazioni possibili sono le più disparate, con una notevole varietà di sensori disponibile sul mercato al giorno d'oggi.

Nel prossimo capitolo verrà descritta ed analizzata la strumentazione hardware necessaria per lo sviluppo del nostro caso specifico di applicazione, il monitoraggio di temperatura, umidità e rumore.

2 Protocolli di comunicazione

In questo capitolo andremo a descrivere i principali protocolli di comunicazione, in particolare quelli previsti per l'utilizzo dal **Meshlium**, gateway di *Libelium* di cui discuteremo più approfonditamente nel capitolo successivo, utilizzando tecnologia wireless a basso consumo.

2.1 802.15.4

IEEE 802.15.4 è uno standard che definisce il funzionamento di una rete personale wireless a bassa velocità (*LR-WPAN*). È adottato da diversi standard, come *ZigBee*, *WirelessHart* e altri. 802.15.4 va a specificare il *livello fisico* e il *livello MAC* (Media Access Control) per i LR-WPAN.

Il livello fisico è più basso del modello *ISO-OSI*, con i protocolli che trasmettono i pacchetti che verranno utilizzati.

Il MAC abilita la trasmissione di frame attraverso il canale fisico, compresi i controlli sulla convalida dei frame e sugli slot temporali. Dato che i collegamenti a bassa potenza rischiano di essere insabili, al livello MAC possiamo pensare di applicare alcune tecniche che ci garantiscono la corretta comunicazione:

- **Low Power Listening - LPL:** ogni volta che un nodo viene invocato («wake up»), si appoggia sulla radio e cerca attività. Se rileva un'attività, il nodo rimane in ascolto per i pacchetti in arrivo. Dopo averli ricevuti, si ferma e torna in «sleep». Questa è una buona soluzione per non drenare l'energia, svegliando i nodi solo quando necessario.
- **Time Slotted Channel Hopping - TSCH:** l'assegnazione degli slot è basato sul tempo e sul canale di frequenza, in questo modo garantiamo parallelismo facendo sì che diverse trasmissioni su diversi canali non interferiscono. Inoltre, se un canale ha interferenze garantiamo la sicurezza delle altre trasmissioni parallele.

2.2 Bluetooth Low Energy - BLE 4.0

Bluetooth permette la comunicazione wireless tra dispositivi eterogenei, come smartphone, PC, stampanti, ecc. In particolare per questo ambito del IOT abbiamo da considerare *Bluetooth Low Energy* e *Bluetooth 4.0*. Utilizzati per sistemi di posizionamento, quali *Morgan*, che permette lo sfruttamento di questa tecnologia in zone affollate.

Un'altra grande differenza sta nella scoperta del dispositivo, il Bluetooth tradizionale richiede l'accoppiamento ed è progettato per la comunicazione uno a uno. **BLE**, invece, è progettato per la comunicazione uno a molti senza che l'associazione sia necessaria.

2.3 Low Power-Wide Area Network - LP-WAN

LP-WAN è una tecnologia che introduce nuovi trade-offs (compromessi), in particolare per ridurre il consumo energetico estendendo il raggio di trasmissione in una via di mezzo tra trasmissione a corto raggio (802.15.4, BLE) e cellulare (GSM, 3G, LTE). Diffonde l'energia di segnale su un'ampia banda di frequenza utilizzando diversi approcci: Spettro di diffusione della sequenza diretta (DSSS), spettro di diffusione del chip (CSS) e Narrow e Ultra-Narrow Band (UNB).

Diversi protocolli si basano su questa tecnologia: *LoRa*, *SigFox*, *NB-IoT*.

2.3.1 Sigfox

SigFox nasce con l'obiettivo di connettere ogni oggetto nel nostro mondo fisico all'universo digitale, costruendo una rete globale dedicata a Internet of Things basato su bassa potenza, lungo raggio e small data che permettono connettività end-to-end. SigFox utilizza la modulazione UNB, il che significa che ogni messaggio è largo 100Hz ed è trasferito a 100-600 bit al secondo.

La vera particolarità di questo protocollo è che un dispositivo non è collegato a una stazione specifica a differenza delle reti cellulari, ma ogni volta che un dispositivo deve inviare un messaggio, trasmetterà il messaggio via broadcast a tutte le stazioni disponibili in quel range, che poi lo elabora e decide se scartarlo o analizzare e inviare il messaggio. Questo rende difficile però l'utilizzo in zone affollate, in cui sono disponibili migliaia di dispositivi.

2.3.2 Nb-Iot

Nb-IoT è uno standard LPWAN standard che usa una narrow band (180-200 kHz); questa tecnologia può funzionare in tre modi diversi: banda GSM standalone, banda LTE in banda o banda di guardia LTE. I dispositivi possono rimanere inattivi e disconnessi fino a quando non devono trasmettere, risparmiando molta energia. Anche in questo caso però, la scalabilità potrebbe rivelarsi una difficoltà, infatti in un contesto dove c'è un gran numero di dispositivi e un numero fisso di disponibili sottoportanti da assegnare, le risorse non sono garantite nemmeno con procedure di accesso casuale.

2.3.3 LoRa

E' una modulazione radio fisica proprietaria, basata sulle tecniche di CSS (chirp spread spectrum), il protocollo MAC invece è LoRaWAN, che è pubblico. LoRaWAN include una larghezza di banda scalabile (125, 250, o 500 kHz) e il segnale ha un costante aumento e decremento della frequenza per codificare un simbolo chiamato *chirp*, una sequenza di bit che fa da principale elemento per la comunicazione.

2.3.4 Zigbee

Zigbee nasce per essere una soluzione economica nel campo del basso consumo in IOT ed è una delle tecnologie più adottate, con dispositivi definiti per ruoli:

- **ZigBee Coordinator (ZC)**: all'interno di una rete può esistere solo uno, può operare come ponte tra più reti ed è in grado di mantenere informazioni riguardanti la rete.

- **ZigBee Router (ZR):** fungono da intermediario per inviare dati ad altri dispositivi nello scambio delle informazioni.
- **ZigBee End Device (ZED):** includono le funzionalità per dialogare col nodo padre, che può essere il coordinator o anche uno dei router.

L'obiettivo di ZigBee è di definire una Wireless mesh network (a maglia), economica e autogestita che possa essere utilizzata per scopi quali il controllo industriale, le reti di sensori, domotica con il più basso consumo energetico.

I dispositivi ZigBee rispettano lo standard *IEEE 802.15.4-2003*, che prevede accesso al canale con modalità *Carrier Sense Multiple Access/Collision Avoidance* (CSMA/-CA); ovvero il nodo, si mette in ascolto del canale per un predeterminato periodo di tempo, e solo se è libero trasmette, altrimenti attende finché il canale non torna inattivo. ZigBee è una delle tecnologie più comuni nel campo della domotica, basti pensare che l'*Echo Plus* di Amazon usufruisce di questa per l'interazione con luci, lampadine ed interruttori. [10]

2.4 Conclusioni

Valutando l'utilizzo di una LP-WAN per risparmio di costi ed energia, Zigbee sembra essere la soluzione più adatta al nostro scopo. Essendo una rete a maglia, tutti i dispositivi connessi a ZigBee possono operare autonomamente, in quanto ciascuno di essi può servire da ripetitore, passando il segnale sugli altri dispositivi.

Questo rende i nodi adottati più versatili, oltre ad essere un protocollo già compatibile con strumenti di domotica di grandi marchi, come Amazon, Philips, Ikea e Xiaomi. Andremo allora ad applicare questo protocollo con il modulo XBee 868LP, descritto nel capitolo seguente insieme al resto della strumentazione hardware adottata.

3 Monitorare rumore, umidità e temperatura

In questo capitolo sarà descritto il sistema sviluppato, partendo dall'hardware. Sarà trattata ciascuna componente in modo individuale, vedendo le motivazioni che ci hanno guidato nella scelta delle stesse.

A seguire, lato software andremo ad esaminare il sistema adottato per la raccolta e la visualizzazione dei dati. Infine, nel sottocapitolo del sistema proposto verrà presentato il sistema nel suo insieme, andando a focalizzarci sull'interazione tra componenti.

3.1 Hardware

Avendo fatto le dovute introduzioni sul IOT, l'utilizzo in Smart Cities e sui protocolli e le tecnologie adottate per la comunicazione, in questa sezione verrà descritto l'aspetto hardware del sistema proposto, dando uno sguardo nel dettaglio di ciascuna componente.

3.1.1 Wasmote v15

Il nostro obiettivo è quello di ideare un sistema in IoT per smart cities in grado di monitorare rumore, umidità e temperatura. Il primo strumento di cui saremo dotati è la sensor board **Wasmote v15** di *Libelium*. La scheda Wasmote è un dispositivo modulare che ci permette di installare sensori diversi e diversi ricetrasmittenti radiofoniche.

L'architettura hardware di Wasmote è stata progettata per essere di basso consumo (0.7 uA in hibernate mode, permettendo così di risparmiare batteria quando non trasmette). E' una board molto flessibile, infatti permette a una grande varietà di sensori di connettersi: CO, CO2, pressione, umidità, pH, luminosità e così via. Sono disponibili 15 diverse interfacce wireless per Wasmote, comprese quelle a lungo raggio (4G / NB-IoT / Cat-M / LoRaWAN / LoRa / Sigfox / 868 MHz / 900 MHz), a medio raggio (ZigBee / 802.15.4 / DigiMesh / WiFi) e a corto raggio (RFID-NFC / Bluetooth 2.1 / Bluetooth Low Energy). [11]

Avere in dotazione un suo IDE open source (API libraries e compiler) ne facilita l'utilizzo anche a chi è alle prime esperienze di sviluppo IoT.

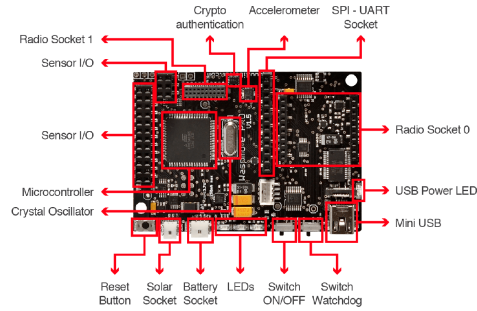


Figura 3.1: Il Wasmote nel dettaglio

E' inoltre pensato per la programmazione *Over the Air* (OTAP), che permette l'aggiornamento software di dispositivi (principalmente è usato per dispositivi mobili come i telefoni cellulari) in rete via connessione wireless, interrogando il server FTP designato per conoscere una nuova versione del software.

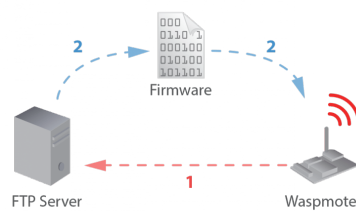


Figura 3.2: Aggiornamento in rete wireless sfruttando OTAP

3.1.2 Wasmote smart cities PRO

Lo scopo della scheda **Wasmote Smart Cities PRO** è quello di estendere le funzionalità di monitoraggio da ambienti interni a luoghi esterni, al fine di realizzare progetti IoT in Smart Cities e ambienti urbani. Questo tipo di board presenta socket 1,2,3,4,5 e 6 per i principali sensori e socket A,B,C,E ed F per i tipi di sensori di tipologia «plug & sense» di Libelium. Posizioneremo questa board sul precedente Wasmote v15 inserendolo attraverso i socket sottostanti.

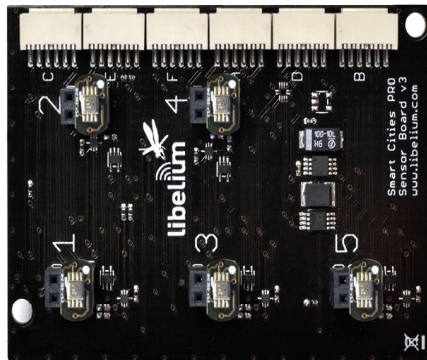


Figura 3.3: Smart cities PRO sensor board

Sul socket 1, andiamo a innestare il sensore **BME280** per individuare *temperatura*, *umidità* e *pressione*. Il sensore di umidità fornisce un tempo di risposta rapido, assicurando un'elevata precisione complessiva su un ampio intervallo di temperature, restituite in gradi C celsius. Il sensore di pressione restituisce invece i dati espressi in percentuale Rh (relative humidity), l'umidità invece viene data in percentuale, con un accuratezza di $\pm 3\%$.^[12]

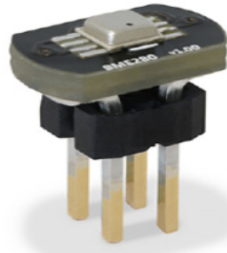


Figura 3.4: Sensore BME280

Al socket A colleghiamo invece il **noise level sensor** con un adattatore probe OEM. Il dispositivo prevede un socket a cui va collegato il microfono che utilizzeremo per captare il suono, uno per il data cable di un eventuale dispositivo Plug & Sense Smart Cities PRO e il connettore dell'alimentazione elettrica.

I dati raccolti sono in dBA (decibel pesato A), il range riconosciuto dal sensore è da 50 dBA a 100 dBA.

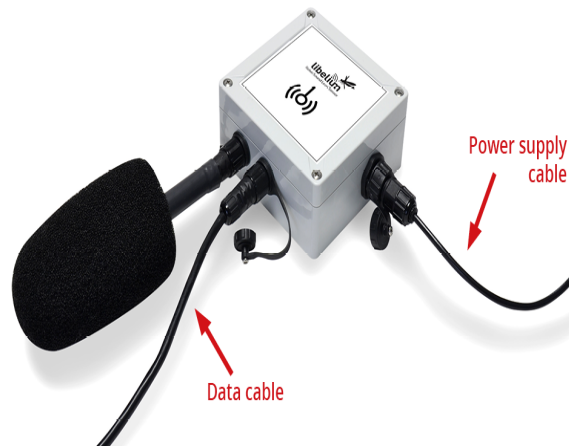


Figura 3.5: Noise sensor level

3.1.3 Modulo XBee 868LP

Un modulo serve a consentire la comunicazione wireless tra il dispositivo e la rete, ed è quindi cruciale per l'IoT. **XBee 868LP** consente la connessione via wireless ai dispositivi end-point nella rete. L'aumento della durata della batteria dei dispositivi IoT è uno degli obiettivi chiave delle tecnologie *LPWAN* (rete a bassa potenza e ampio raggio). Per questo motivo, le caratteristiche di risparmio energetico sono una parte essenziale delle tecnologie cellulari LPWAN, e XBee 868LP è un modulo ad alta performance per basso consumo di batteria. In particolare, nel nostro caso adotteremo i protocolli Zigbee, descritti nel capitolo precedente.

Il modulo XBee 868LP è provvisto di un'antenna a 4.5 dBi, la frequenza utilizzata è la banda 868 MHz, che concede la selezione tra 30 canali via software. I canali sono distanziati di 100kHz l'uno dall'altro. La velocità di trasmissione è di 10 kbps. Questo modulo può essere collegato sia sulla SOCKET0 che la SOCKET1 della Wasp mote board.[13]

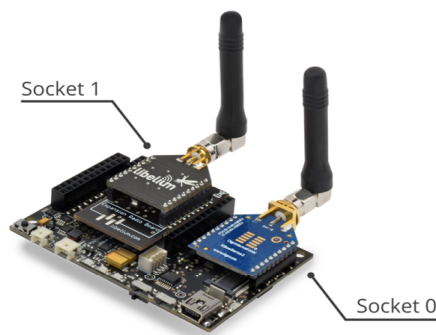


Figura 3.6: Applicazione su Socket0 e Socket1 del Modulo

3.1.4 Meshlium

Il **meshlium** è un gateway di IoT che può contenere fino a 4 diverse interfacce radio: un WiFi 2.4 GHz (Access Point), un 4G/3G/GPRS/GSM e 2 radio XBee/RF. Il meshlium può includere anche radio Bluetooth, BLE e WiFi per le applicazioni di scansione. L'involucro in alluminio IP67 consente ad un meshlium di essere posizionato all'aperto. Il meshlium può essere adoperato come router da RF (XBee) a ethernet/4G/3G/GPRS/GSM per i nodi Waspote, come WiFi AP (access point), router da WiFi a 4G/3G/GPRS/GSM, real-time tracker da GPS a 4G/3G/GPRS/GSM, oppure scanner di smartphone (in grado di rilevare dispositivi iPhone/Android).

Tutte le opzioni di rete possono essere controllate dal Manager System, un'interfaccia web fornita con Meshlium. Permette di controllare le interfacce e le opzioni di sistema in modo sicuro (l'accesso al Manager System prevede un login, inoltre è possibile visionare tutti i devices connessi), facile e veloce.

Il meshlium è provvisto di un hard disk di 16 GB, di cui 3 dedicati al Manager System.

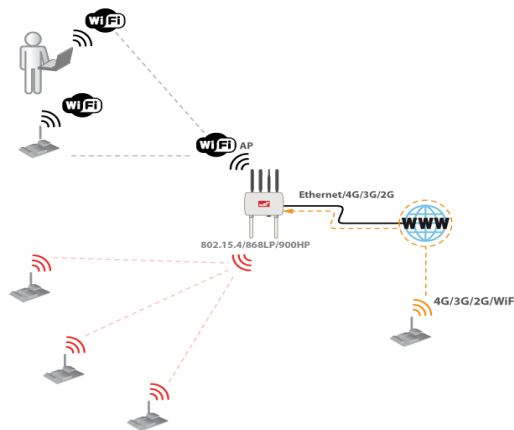


Figura 3.7: Meshlium WiFi AP

Il meshlium è provvisto di sei connettori per le sue antenne, un connettore ethernet e un connettore per nanoSIM/microUSB.

Le antenne utilizzate, contrassegnate nella figura in basso, sono:

- a) Dipole 5 dBi (per Bluetooth, BLE, WiFi, XBee-PRO 802.15.4).
- b) Antenne per 4G / GPS.
- c) Dipole 4.5 dBi (per XBee 868LP, XBee-PRO 900HP).

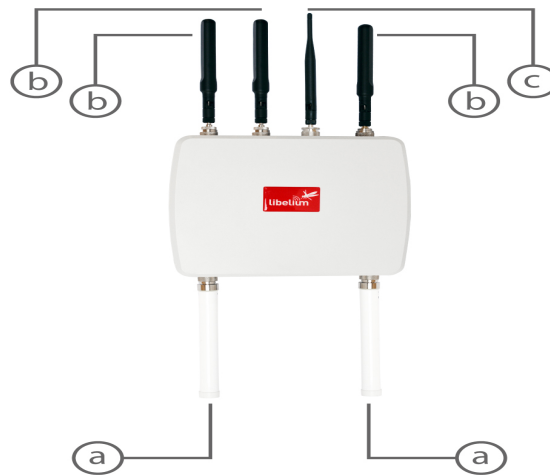


Figura 3.8: Le antenne del Meshlium

La connessione tra meshlium e computer viene effettuata attraverso lo **switch** in dotazione, che presenta connettori LAN, POE, e DC.

Lo switch è un dispositivo di rete che si occupa della comunicazione a livello 2 del modello ISO/OSI (*datalink*).

Come è possibile vedere in Figura 3.9, esso è composto da tre porte:

- **LAN** va connesso al pc in utilizzo, con cavo crossover UTP, così da effettuare lo scambio di dati con il gateway.
- **POE** (acronimo di *Power over Ethernet*) consente al cavo Ethernet FTP di fornire alimentazione ai dispositivi di rete tramite la connessione dati ottenuta

dal pc.

- **DC** (*Direct Current*, corrente alternata) è dedicato all'alimentatore del Meshlium.

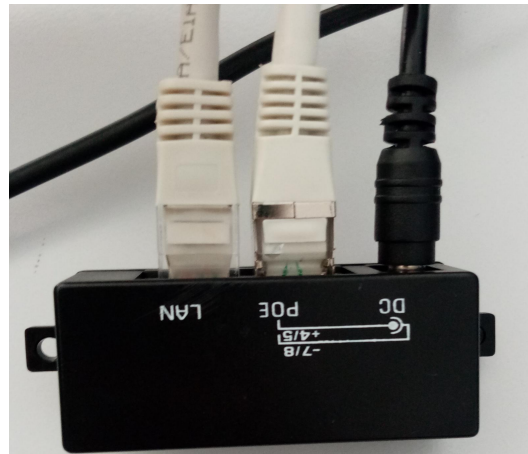


Figura 3.9: Switch

3.2 Software

Abbiamo fissato i concetti e le definizioni alla base dell'utilizzo di moduli, gateway e le board principali di riferimento. Ora che è stato sviscerato tutto il lato hardware dei componenti necessari alla prosecuzione del nostro progetto, possiamo mostrare lo sviluppo lato software per l'obiettivo preposto, dagli strumenti messi a disposizione da Libelium allo sviluppo di una pagina web per la visualizzazione dei dati raccolti.

3.2.1 Meshlium Manager System

Attraverso il *Manager System* proprietario di Libelium è possibile accedere e impostare il gateway Meshlium. All'indirizzo 10.10.10.1, utilizzando un browser web è possibile visualizzare la pagina del login, in cui vanno inseriti username e password provvisti da Libelium.



Figura 3.10: Login Meshlium Manager System

Una volta effettuato l'accesso, l'applicazione web presenta l'interfaccia come da immagine illustrata subito in basso: **Interfaces** permette di fare setup per ethernet, WiFi Access Point, 4G e il proxy; **Sensor Networks** gestisce i sensori connessi e

dirige le informazioni recepite al database del gateway, interno o esterno che sia, mediante impostazioni presenti in questo spazio; **Cloud Connectors** permette di impostare il Meshlium per permettere una connessione a una piattaforma cloud di terze parti; **Device Connectors** allo stesso modo riguarda le impostazioni per la connessione a device di terze parti.

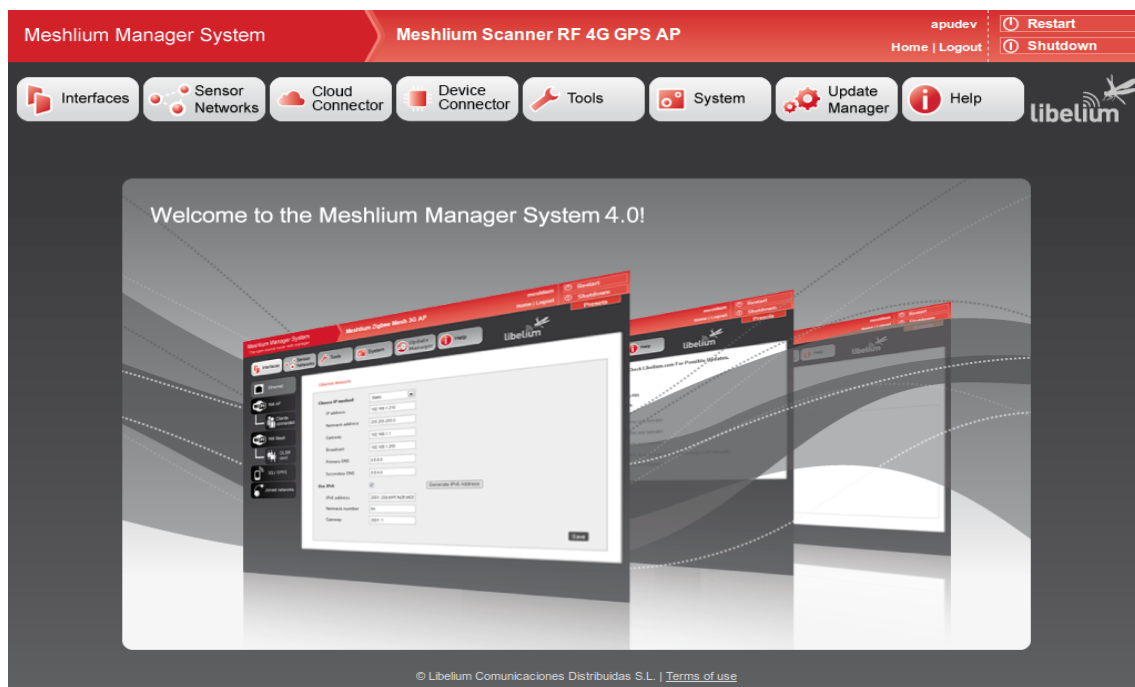


Figura 3.11: Meshlium manager system

3.2.2 MySQL

MySQL è un sistema di gestione di database relazionali (RDBMS) gratuito e open source che utilizza SQL (Structured Query Language). **MySql Workbench** è uno strumento visuale per gestori di database, sviluppatori e DBA (Database administrator). MySQL Workbench fornisce modellazione dei dati, sviluppo SQL e strumenti

di amministrazione per la configurazione di server, l'amministrazione degli utenti e il backup dei dati.

3.3 Sistema proposto

In questa sezione viene spiegata l'architettura del sistema, a partire dalla fase di Sensing, fino alla visualizzazione dei dati. L'architettura di riferimento è quella rappresentata in Figura 3.12. Divideremo la presentazione del sistema in quattro fasi: Sensing, Trasmissione, Salvataggio e Visualizzazione.

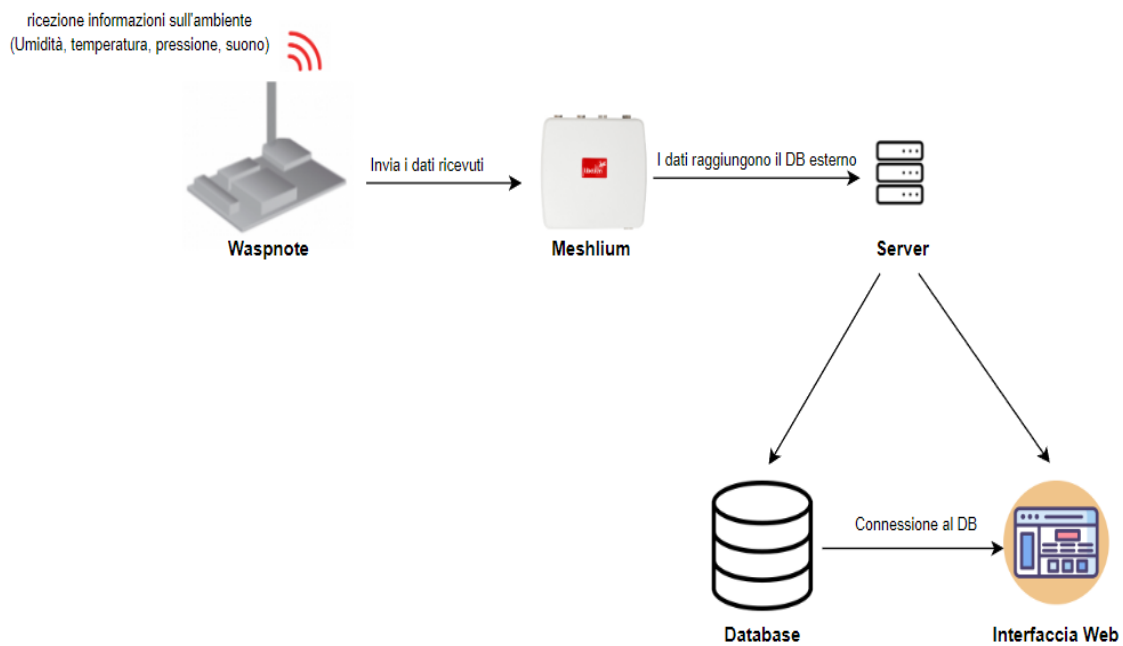


Figura 3.12: Diagramma sistema proposto

3.3.1 Sensing

La programmazione del *BME 280* permetterà la ricezione di umidità, pressione e temperatura, invece il *noise level sensor* verrà programmato, come suggerisce il nome, per leggere i valori del rumore nell'ambiente designato. Mediante il modulo *XBee 868LP*, la board sarà in grado di comunicare i dati al gateway.

```
#include <WaspXBee868LP.h>

// PAN (Personal Area Network) Identifier
uint8_t panID[2] = {0x12,0x34};

// Define Freq Channel to be set:
uint8_t mask[4] = {0x1F,0xFF,0xFF,0xFF};

// Define preamble ID
uint8_t preambleID = 0x06;

// Define the Encryption mode: 1 (enabled) or 0 (disabled)
uint8_t encryptionMode = 0;

// Define the AES 16-byte Encryption Key
char encryptionKey[] = "WaspmoteLinkKey!";
```

Figura 3.13: Configurazione Xbee

La porzione di codice in figura 3.13 mostra come vanno impostati i parametri del Xbee per comunicare col gateway, con:

- **PAN:** identificatore della Personal Area Network, per la trasmissione di pacchetti.
- **Mask:** definisce il canale che sarà usato per la comunicazione; è necessario trovare un canale libero al fine di evitare collisioni.
- **Preamble ID:** viene usato come estensione al PAN.

Infine, *Encryption mode e key* definiscono la cifratura dei dati trasmessi al gateway. Tali parametri dovranno essere gli stessi utilizzati dal Manager System di Libelium per consentire una corretta comunicazione.

In Figura 3.14 viene mostrato l'inizializzazione necessaria per l'invio di pacchetti. L'operazione avviene semplicemente settando l'identificativo del Waspmote. La fase di invio pacchetti continua andando a definire la tipologia di frame che sarà inviato, nel nostro caso ASCII; e andando ad aggiungere al sensore i campi di cui abbiamo bisogno; nell'esempio di Figura 3.15, sarà inviato un dato relativo alla batteria.


```
void setup()
{
    // init USB port
    USB.ON();
    USB.println(F("Sending packets example"));

    // store Wasmote identifier in EEPROM memory
    frame.setID( WASPMOTE_ID );

    // init XBee
    xbee868LP.ON();
}
```

Figura 3.14: Setup Xbee

```
// create new frame
frame.createFrame(ASCII);

// add frame fields
frame.addSensor(SENSOR_STR, "new_sensor_frame");
frame.addSensor(SENSOR_BAT, PWR.getBatteryLevel());

////////////////////////////////////
// 2. Send packet
////////////////////////////////////

// send XBee packet
error = xbee868LP.send( RX_ADDRESS, frame.buffer, frame.length );
```

Figura 3.15: Invio frame Xbee

3.3.2 Trasmissione

Con la connessione al gateway *Meshlium*, si accede al suo *Manager System*. Nella scheda dedicata ai Sensor Network, vengono presentati i campi del database locale, ma abbiamo la possibilità di usufruire di un database esterno, effettuando la connessione al nostro *MySQL Workbench*.

Vanno allora definite le informazioni adatte al nostro utilizzo, riportate in foto.

Captured Data

Local DataBase External Database Show me NOW Advanced

Connection data

Database: MeshliumDB

Table: sensorParser

IP: 10.10.10.22

Port: 3306

User: utentemesh

Password: [obscured]

☐ Store frames in the external data base Save

Synchronization limit: 100

Show data Last 100 insertions. Show sql script (to create database and table)

Save Check Connection Mark all as synchronized

| ID | Date | ID Wasp | ID Secret | Fr. type | Fr. number | Sensor | Value |
|-------|---------------------|---------|-----------|----------|------------|--------|-------|
| 73568 | 2013-10-10 18:00:01 | N1 | 32980099 | 253 | 233 | BME280 | 40 |
| 73554 | 2013-10-10 17:10:01 | N1 | 32980099 | 253 | 233 | BME280 | 45 |

Figura 3.16: Impostazione database esterno

3.3.3 Salvataggio

Scegliendo di avere un database esterno rispetto al predefinito immagazzinamento sul Meshlium via database locale, impostando adeguatamente il collegamento dal Meshlium Manager System a MySQL, su MySQL il database nominato come «MeshliumDB» conterrà la tabella *Sensor Parser*, che contiene i seguenti campi descritti nella Tabella 3.1.

| <i>Nome</i> | <i>Tipo</i> | <i>Descrizione</i> |
|---------------------|-------------|------------------------------------------------------------------------------------------------------------|
| Id | INT | Identificativo della riga nel DB, primary key |
| IdWaspnote | VARCHAR | Identificativo del waspnote su cui è stato utilizzato il sensore da cui è stata letta l'informazione |
| IdSecret | VARCHAR | Identificativo segreto assegnato in fase di mapping |
| Frame Type | INT | Riconoscitivo della tipologia di frame |
| Frame Number | INT | Numero di frame |
| Sensor | VARCHAR | Il sensore da cui proviene l'informazione |
| Value | VARCHAR | Valore dato ottenuto |
| Time stamp | TIMESTAMP | Data e ora della lettura |
| Sync | BIGINT | |
| Raw | VARCHAR | |
| Parser Type | TINYINT | Tipologia del parser |
| MeshliumID | VARCHAR | Identificativo del gateway che ha raccolto l'informazione |

Tabella 3.1: Campi del DB Sensor Parser

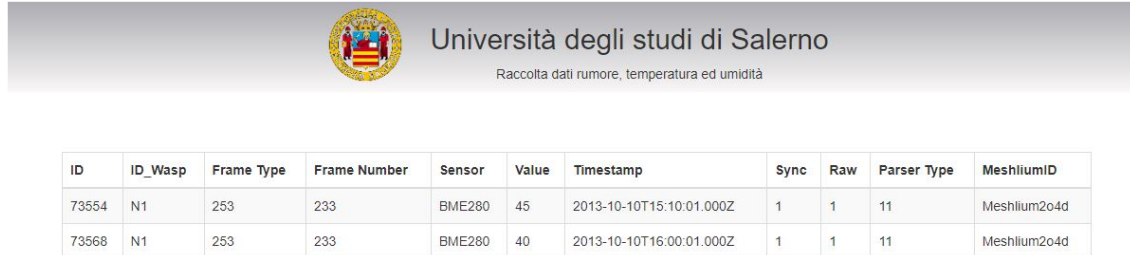
3.3.4 Visualizzazione

Per rendere possibile la visualizzazione dei dati raccolti al client, è stata realizzata una pagina web connessa al database descritto nella sottosezione precedente. Lato server è stato implementato **MySQL**, **Cors**, che permette ad una web application di fare richieste HTTP per protocolli, porte e dominio ad un altro client, **Express**, il framework backend per la creazione di applicazioni web, e infine **Node.js** per il server in javascript. Per il client invece, abbiamo utilizzato **React** per la realizzazione dell'interfaccia utente (quindi frontend), con l'utilizzo di *ReactDOM* per il routing. Implementare **Axios** inoltre permetterà di richiedere la pagina dal web server.

Lo sviluppo del programma è stato effettuato tramite l'editor **Visual Studio Code**, un editor multi-piattaforma il quale, tramite delle estensioni, permette di evidenziare la sintassi di ciascun linguaggio di programmazione e aiutare durante la scrittura. Il progetto è stato organizzato in due sezioni, come descritto precedentemente, quindi abbiamo il lato server con un file *index.js*, in cui viene stabilita la connessione col database MySQL (tramite la funzione «createPool»). Stabilita la connessione col db, andiamo a raccogliere i dati di SensorParser con una select. La select viene allora utilizzata da una get per «app», che abbiamo stabilito essere una costante che utilizza Express. Stabilita una port inoltre, il server ora è in ascolto su quella porta tramite una listen su app.

Lato client abbiamo un file *app.js* in cui andiamo ad utilizzare React per creare una root, che ci permetterà di stabilire la pagina principale dell'interfaccia, descritta in *Home.js*. Qui axios carica i dati ottenuti dal server (e quindi dal db) con una get che permetterà di settare i dati. Fatto questo, andiamo a stabilire cosa viene visualizzato nell'interfaccia web, quindi l'header e la tabella (come da foto in basso), che viene riempita con i campi del database. Le manipolazioni grafiche necessarie sono

descritte da *App.css*.



The screenshot shows the header of a web application for the University of Salerno. It features the university's logo and name, along with the text 'Raccolta dati rumore, temperatura ed umidità'. Below the header is a table with 11 columns: ID, ID_Wasp, Frame Type, Frame Number, Sensor, Value, Timestamp, Sync, Raw, Parser Type, and MeshliumID. The table contains two rows of data.

| ID | ID_Wasp | Frame Type | Frame Number | Sensor | Value | Timestamp | Sync | Raw | Parser Type | MeshliumID |
|-------|---------|------------|--------------|--------|-------|--------------------------|------|-----|-------------|--------------|
| 73554 | N1 | 253 | 233 | BME280 | 45 | 2013-10-10T15:10:01.000Z | 1 | 1 | 11 | Meshlium2o4d |
| 73568 | N1 | 253 | 233 | BME280 | 40 | 2013-10-10T16:00:01.000Z | 1 | 1 | 11 | Meshlium2o4d |

Figura 3.17: Interfaccia web

3.4 Conclusioni

Sono state analizzate e descritte accuratamente tutte le componenti hardware e software necessarie per la riuscita del nostro progetto, e indicando il flusso che porta dalla lettura del dato alla visualizzazione da parte dell'utente, abbiamo ora realizzato la nostra interfaccia web che presenta in tabella i dati raccolti dai sensori applicati.

4 Conclusioni

Approfondendo conoscenze sui reali impieghi attuali del IoT, è stato portato a termine lo sviluppo del sistema di monitoraggio ambientale per la lettura di dati su umidità, temperatura, pressione e rumore in smart cities.

Da un punto di vista prettamente hardware, dato che il sistema proposto presenta un database con campi che differenziano le tipologie di dati ricevuti, viene data la potenzialità di espandere il sistema di monitoraggio con ulteriori sensori compatibili con la sensor board utilizzata.

Valutando la scelta di portare esternamente il database del gateway per sviluppare una nostra interfaccia web rispetto a quella proposta dall'accesso al Manager System di Libelium, abbiamo possibilità futura di aggiornarla, aggiungendo nuove funzionalità visibili all'utente. Ora che il sistema è a tutti gli effetti aperto agli utenti, e quindi non solo a chi ha l'accesso alla gestione del gateway, i dati resi disponibili possono essere utilizzati, come ad esempio mostrato negli esempi di applicazioni IoT, per analizzare dati in modo da prendere decisioni aziendali in modo mirato (data analytics), per sviluppare modelli statistici (machine learning) o scoprire relazioni dai dati estratti (data mining).

Elenco delle figure

| | | |
|------|--------------------------------------------------|----|
| 1.1 | Processi di IoT | 11 |
| 1.2 | Architettura monitoraggio inquinamento | 14 |
| 1.3 | Temperature inquinamento | 15 |
| 1.4 | Notifica inquinamento | 16 |
| 1.5 | Daily Report | 16 |
| 1.6 | Setup lampione fotovoltaico | 17 |
| 1.7 | Analisi dati batteria MatLab | 18 |
| | | |
| 3.1 | Waspmote | 27 |
| 3.2 | OTAP | 27 |
| 3.3 | Smart cities PRO sensor board | 28 |
| 3.4 | BME280 | 29 |
| 3.5 | Noise sensor level | 29 |
| 3.6 | Xbee 868LP | 30 |
| 3.7 | WiFi AP Meshlium | 31 |
| 3.8 | Antenne Meshlium | 32 |
| 3.9 | Switch | 33 |
| 3.10 | Login Meshlium | 34 |
| 3.11 | Meshlium manager system | 35 |

| | | |
|------|-----------------------------------------|----|
| 3.12 | Diagramma sistema proposto | 37 |
| 3.13 | Configurazione Xbee | 38 |
| 3.14 | Pacchetti Xbee | 39 |
| 3.15 | Invio frame Xbee | 39 |
| 3.16 | Impostazione database esterno | 40 |
| 3.17 | Interfaccia web | 43 |

Bibliografia

- [1] Fulvio De Santis. L'internet delle cose – iot e sensori. <https://it.emcelettronica.com/linternet-delle-cose-iot-e-sensori>.
- [2] Ammar Gharaibeh, Mohammad A. Salahuddin, Sayed Jahed Hussini, Abdallah Khreishah, Issa Khalil, Mohsen Guizani, and Ala Al-Fuqaha. Smart cities: A survey on data management, security, and enabling technologies. *IEEE Communications Surveys & Tutorials*, 19(4):2456–2501, 2017.
- [3] Cities directory. <https://www.firmware.org/about-us/smart-cities/cities-directory/>.
- [4] Alessandra Gualtieri. Smart city nel mondo: Impariamo dai migliori. <https://www.pmi.it/pubblica-amministrazione/riforma-pa/360018/smart-city-nel-mondo-impariamo-dai-migliori.html>, Jun 2021.
- [5] Rong Du, Paolo Santi, Ming Xiao, Athanasios V. Vasilakos, and Carlo Fischione. The sensible city: A survey on the deployment and management for smart city monitoring. *IEEE Communications Surveys & Tutorials*, 21(2):1533–1560, 2019.
- [6] Ameer A. Al-Shammaa and A. J. Stocker. Discovering neighbour nodes based on signal strength using waspmote nodes. In *2019 IEEE SENSORS*, pages 1–4,

- 2019.
- [7] Baihaqi Siregar, Azmi Nur Nasution, and Dedy Arisandi. Air pollution monitoring system using waspmote gases sensor board in wireless sensor network. In *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, pages 83–88, 2020.
- [8] M. S. B. Shah Rizam, Md. Tahir Nooritawati, J. Haffiz, and M. Y. Yusnani. Solar streetlight remote monitoring system using waspmote wireless sensor network module (wsn) for system efficiency enhancement. In *2014 IEEE Conference on Systems, Process and Control (ICSPC 2014)*, pages 153–156, 2014.
- [9] Yusnani Mohd Yusoff, Roszainiza Rosli, Mohd Uzir Karnaluddin, and Mustaffa Samad. Towards smart street lighting system in malaysia. In *2013 IEEE Symposium on Wireless Technology & Applications (ISWTA)*, pages 301–305, 2013.
- [10] Lorenzo Ricciutelli. Zigbee: Cos'è e come funziona. https://www.domoticafull.it/zigbee-cose-e-come-funziona/#Che_cos8217e_e_a_cosa_serve, Sep 2022.
- [11] Wasmote the smart sensor platform to develop iot projects. <https://www.libelium.com/iot-products/wasmote/>, Jul 2022.
- [12] Bme280 sensor breakout - adafruit. <https://www.digikey.it/it/pdf/a/adafruit/bme280-sensor-breakout>.
- [13] Libelium xbee 868 lp. <https://development.libelium.com/868-lp-networking-guide/hardware>.

Dediche e ringraziamenti

A conclusione di questo elaborato, desidero menzionare le persone senza le quali non avrei avuto il supporto necessario a compiere questi tre anni di studi.

Ringrazio il mio relatore, il prof. Christiancarmine Esposito, che in questi lunghi mesi ha saputo guidarmi nel percorso di tirocinio coronato con la tesi e nelle ricerche nelle conoscenze necessarie dei temi trattati.

Ringrazio i miei colleghi, Maria Rosaria, Carmine, Alessio e Alessandro, per aver condiviso con me l'esperienza universitaria e per il supporto tecnico. Sia che stiate continuando gli studi o affacciandovi al mondo del lavoro, sono sicuro che brillerete.

Ringrazio i miei fratelli, non di sangue ma di certo di cuore, Antonio, Ivan, Federico e Gerardo, per essere miei compagni affezionati da più di un decennio. Chiamarvi amici, migliori amici, è un mio grande e sentito onore.

Ringrazio anche l'amico Davide, per l'affetto e per la sua sempre gradita saggezza e pazienza.

Ringrazio la mia ragazza, Carmen, per essere da più di un anno mia confidente e di essere sempre di sostegno nei momenti del bisogno.

Ringrazio i miei genitori per avermi permesso di portare a termine in serenità il percorso di studi, per tutti i loro sforzi e sacrifici.