UNIVERSITÀ
DEGLI STUDI
DI PALERMO

NDS
NETWORKING AND DISTRIBUTED SYSTEMS
RESEARCH GROUP

Gestione di dispositivi wearable per l'health monitoring

Tesi di Laurea Magistrale in Ingegneria Informatica

Claudio Sorrenti

Relatore: Prof. Daniele Peri

UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO DI INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

GESTIONE DI DISPOSITIVI WEARABLE PER L'HEALTH MONITORING

TESI DI LAUREA DI
CLAUDIO SORRENTI

RELATORE
Prof. DANIELE PERI

CONTRORELATORE
Prof.ssa LILIANA LO PRESTI

ANNO ACCADEMICO 2019 – 2020

Indice

Premessa

Negli ultimi anni il tema del monitoraggio della salute (health monitoring) sta rivestendo un ruolo importante in ambito tecnologico, suscitando un interesse sempre più crescente da parte della popolazione. A tal propositivo sono stati sviluppati dispositivi indossabili (wearable) in grado di monitorare alcuni parametri clinici, basti pensare ai più popolari smartwatch che possono registrare il battito cardiaco e l'ossigenazione del sangue. Parallelamente sono nate le cosiddette smart home, costituite da una rete di dispositivi fissi che vengono collocati in diversi punti all'interno dell'abitazione. Tali dispositivi possono monitorare i parametri ambientali oppure offrire servizi agli utenti al fine di migliorare la qualità della vita sia in termini di comfort che di sicurezza.

L'obiettivo di questa tesi è quello di proporre un sistema di smart environment orientato all'health monitoring, in cui dispositivi indossabili (wearable) e fissi cooperano e scambiano informazioni circa lo stato di salute del paziente all'interno della sua abitazione. Il sistema in esame è stato pensato per quelle persone che, per motivi di anzianità e/o a causa di determinate patologie, necessitano di specifica assistenza e allo stesso tempo vivono sole in casa.

Come da titolo, questa tesi, suddivisa in cinque capitoli, si concentrerà sul lato wearable del sistema. Nel primo capitolo vedremo una panoramica sulla tecnologia wearable con specifico riferimento all'health monitoring e ai dispositivi attualmente presenti in commercio, nel secondo, invece, saranno introdotti gli smart environment, le applicazioni distribuite e i principali protocolli di comunicazioni adottati in questi contesti. Nei capitoli tre e quattro sarà invece illustrata la parte sperimentale su cui si è scelto di focalizzare il lavoro. In particolare, nel capitolo tre verrà analizzato

un caso di studio circa il problema del rilevamento delle cadute illustrando lo sviluppo di un'applicazione a supporto di tale tematica. Nel capitolo quattro, inoltre, sarà illustrato uno strumento di testing di applicazioni distribuite basato sulla virtualizzazione di sensori, in cui vengono utilizzati i dati contenuti in un dataset per simulare tutti i test desiderati. Nello specifico si vedrà un esperimento in cui un dispositivo, che interpreta il ruolo di nodo fisso, virtualizza un accelerometro comunicando via Bluetooth con un altro dispositivo, che interpreta il nodo wearable. Infine, nel capitolo conclusivo saranno esposte tutte le considerazioni sui risultati ottenuti, al fine di valutare le performance dello strumento proposto e i possibili sviluppi futuri in un'ottica più ampia.

1 Introduzione alla tecnologia wearable

Al giorno d'oggi il progresso tecnologico sta portando alla realizzazione di calcolatori sempre più piccoli in grado di eseguire complesse operazioni di calcolo. Le esigue dimensioni di tali dispositivi permettono non solo di risolvere i problemi di ingombro legati al trasporto, come poteva accadere fino a non molto tempo fa, ma anche di permettere agli utenti di usufruire dei servizi tecnologici odierni ovunque essi si trovino. Basti pensare agli smartphone che oggi possono essere ritenuti a tutti gli effetti dei personal computer tascabili capaci di offrire una grande varietà di servizi più o meno complessi, come la navigazione web, la posta elettronica, le web chat, i social networks e tanti altri. Tale progresso ha portato anche alla nascita di dispositivi che possono addirittura essere indossati, riducendo al minimo i problemi legati al trasporto.

Un dispositivo indossabile (wearable) può essere definito come un piccolo calcolatore dalle dimensioni e peso ridotte in grado di offrire una serie di funzionalità smart all'utente che lo indossa. Lo scopo di tali dispositivi è quello di migliorare la vita quotidiana degli individui che li indossano, rendendo quanto più possibile fruibile e meno invasiva la tecnologia al servizio dell'utente e offrendo servizi aggiuntivi in grado di integrare le percezioni ambientali con quelle sensoriali umane, talvolta amplificandole.

In generale un dispositivo del genere è composto da un microcontrollore, un set di sensori, che possono variare in base ai parametri o agli eventi che si desidera monitorare, e infine da un modulo di comunicazione wireless per interagire con un dispositivo di supporto, comunemente uno smartphone.

Questi dispositivi, in quanto wearable, possono assumere forme piuttosto varie a seconda dell'utilizzo a cui sono destinati. I più popolari si ispirano ai classici orologi da polso, ereditandone la comodità e l'eleganza estetica, altri invece sono nascosti all'interno di accessori che indossiamo quotidianamente come magliette, scarpe, cappelli, occhiali, ecc. I campi applicativi sono tra i più vari e spaziano dall'entertainment, all'uso sportivo, al monitoraggio della salute. Volendo citare qualche esempio, nel mercato odierno spiccano sicuramente gli smartwatch, i quali, oltre ad indicare l'ora, offrono servizi di notifica, sveglia, cronometro, activity tracking, sleep tracking e molto altro. In commercio sono disponibili anche scarpe smart, che forniscono informazioni circa la postura di chi le indossa, e magliette smart, che monitorano i movimenti dell'individuo durante una sessione di allenamento.

Generalmente, tutti questi dispositivi operano individualmente, appoggiandosi di solito ad una app su smartphone che ne permette la gestione completa, con la possibilità di programmare tutte le relative funzionalità. L'idea, quindi, è quella di utilizzare il proprio smartphone per controllare tutti i dispositivi wearable che si possiedono e tale modello è sicuramente valido quando si gestiscono pochi dispositivi. In alcuni contesti come, ad esempio, quello domestico, potrebbero anche essere presenti numerosi dispositivi fissi oltre che wearable, per cui sarebbe interessante che essi condividessero all'interno di un'unica infrastruttura di rete le informazioni riguardanti sia il soggetto sia l'abitazione in cui egli vive.

A tal proposito, l'idea che si intende proporre in questa tesi è quella di un dispositivo wearable per health monitoring che si integri con un'infrastruttura fissa al fine di scambiare informazioni circa lo stato di salute di un paziente. Solitamente, tali individui sono persone anziane, per cui l'idea di un dispositivo wearable la cui gestione è vincolata all'utilizzo tramite smartphone potrebbe risultare poco pratica.

Vedremo più avanti tutti i dettagli del sistema proposto, adesso seguirà un'introduzione al mondo della tecnologia wearable, partendo dalle origini fino ad arrivare ai giorni nostri e alla sua applicazione in ambito di health monitoring.

1.1 Storia

I primi dispositivi che vedremo sono poco conciliabili con il concetto di "wearable" se paragonati a quelli moderni, ma vale la pena citarli dal momento che vengono considerati i precursori di questa famiglia di dispositivi. Partiamo da quello che può essere considerato il primo della storia, fino ad arrivare a quelli presenti nel mercato odierno.

1.1.1 L'antenato dei dispositivi wearable

Il primo dispositivo che si possa definire wearable risale agli anni '60 e viene attribuito a Claude Shannon ed Edward Thorp [1]. I due matematici unirono le loro conoscenze scientifiche e la loro passione per il gioco d'azzardo per realizzare un dispositivo che gli permettesse di barare alla roulette. Comprarono una roulette e iniziarono i loro esperimenti studiandone la fisica, curve e traiettorie della pallina, tempi e velocità di rotazione del piatto. Misero così a punto un algoritmo capace di aumentare di circa il 44% le probabilità di vincita. A questo punto costruirono un piccolo computer dalle dimensioni di un pacchetto di sigarette e lo programmarono secondo l'algoritmo da essi concepito. Fu costruito

Figura 1.1 - Il dispositivo costruito da Shannon e Thorp

appositamente così piccolo al fine di poter essere indossato e soprattutto nascosto affinché non venissero scoperti mentre baravano. Più precisamente, il dispositivo fu nascosto dentro una scarpa ed era collegato con un sottilissimo filo ad un apparecchio acustico.

Ad ogni giro di ruota il microcomputer veniva comandato attraverso l'alluce in modo che memorizzasse i tempi di passaggio della pallina su determinati punti di riferimento. Con questi dati il dispositivo restituiva, sotto forma di nota musicale, il settore della ruota nel quale, con buona probabilità, la sfera si sarebbe fermata. L'esperimento dei due matematici sembrò riuscire davvero bene, al punto che nel 1985 il Nevada lo dichiarò illegale.

Nonostante ciò, l'utilizzo di tale dispositivo era tutt'altro che semplice e comodo. Lo stesso Thorp dichiarò che era necessaria una lunga preparazione per indossare e calibrare l'apparecchio al fine di potersi muovere senza farsi scoprire e senza danneggiare il dispositivo, senza considerare l'immensa scomodità nell'indossarlo.

1.1.2 EyeTap by Steve Mann

Il padre della tecnologia wearable viene sicuramente ritenuto Steve Man [2], che negli anni '80 iniziò a lavorare alla realizzazione di diversi dispositivi indossabili. Tra questi il più famoso è quello che nel 1994 diventerà la Wearable Wireless Webcam, ovvero un sistema composto da una webcam posta davanti all'occhio destro dell'individuo, montata su un caschetto da motociclista e collegata ad un sistema di controllo dalla forma di uno zaino. L'idea era quella di acquisire ciò che l'utente vedeva e di trasmetterlo in real-time sul web [3].

Più avanti questa idea si trasformò nel progetto EyeTap che prevedeva di arricchire la visione dell'utente con una serie di informazioni grafiche elaborate da un computer centrale. Il progetto è stato portato avanti dallo stesso Mann fino ai giorni nostri, il quale, sfruttando la moderna tecnologia, realizzò un dispositivo molto più snello, comodo e ricco di funzionalità rispetto al suo primo prototipo primordiale.

Figura 1.2 - Steve Mann con indosso EyeTap

Con EyeTap, Mann non ha soltanto dato inizio alla tecnologia wearable moderna ma anche a ciò che prenderà il nome di realtà aumentata. Quest'ultima verrà successivamente rivisitata e arricchita da Google con i suoi Google Glass, che costituiscono a tutti l'evoluzione di EyeTap.

1.1.3 Google Glass – Wearable and AR

Tra i dispositivi wearable più recenti e sicuramente più innovativi abbiamo i Google Glass [4], gli occhiali a realtà aumentata sviluppati da Google e lanciati sul mercato nel 2014 con lo scopo di cambiare drasticamente la vita quotidiana degli utenti che li indossano. Le funzionalità offerte da questi occhiali "smart" sono davvero tante: controllo dei social network, consultazione di siti web e notizie online, visualizzazione di mappe e

Figura 1.3 - Google Glass

indicazioni stradali, videoconferenze, foto, video, lettura ed invio di messaggi.

Nonostante l'intenzione di rivoluzionare completamente l'utilizzo di internet e dei suoi servizi, i Google Glass non sembrano avere avuto grande successo, almeno nel mercato di massa, motivo per il quale a gennaio del 2016 il progetto è stato chiuso. Il motivo principale risiede nell'elevato costo del dispositivo, ragion per cui soltanto poche persone possono permettersi il lusso di acquistare tale dispositivo. L'unica area in cui gli occhiali smart di Google hanno avuto successo è quella aziendale/lavorativa, diventando così un'esclusiva per professionisti. Infatti, negli ultimi anni, aziende come GE Aviation (e molte altre) hanno adottato l'utilizzo dei Google Glass, consentendo agli operai di svolgere parte del proprio lavoro con le mani completamente libere [5]. Le istruzioni, infatti, appaiono sul display del dispositivo senza la necessità di dover consultare tablet o manuali. A tal proposito nel 2017 sono stati lanciati i Google Glass Enterprise Edition all'interno del progetto "Glass at Work", stavolta riservati esclusivamente al settore lavorativo-professionale.

1.1.4 Smartwatch e fitness tracker

Tra i principali esempi di dispositivi wearable di maggiore successo degli ultimi anni ci sono sicuramente gli *smartwatch*, ovvero l'evoluzione dei classici orologi da polso digitali. L'origine di questi dispositivi viene comunemente fissata nel 2012 quando la Pebble Technology Corporation realizza e commercializza il Pebble [6], un orologio

Figura 1.4 - Pebble Smartwatch

digitale capace di visualizzare le notifiche del proprio smartphone. I punti di forza di questo prodotto sono stati il prezzo accessibile di circa 100$ e la sua leggerezza (solo 38g), al punto da raggiungere un elevato numero di vendite in pochissimo tempo.

Poco dopo anche tantissime aziende leader nel settore degli smartphone si insediarono nel mercato emergente introducendo la propria idea di smartwatch, basti pensare a Samsung con la sua serie Gear o ad Apple con i suoi Watch. Tale mercato è cresciuto esponenzialmente negli ultimi anni, al punto che nel 2018 sono state stimate 41,5 unità vendute e se ne ipotizzano 81 milioni per il 2021 [3].

Figura 1.5 - Samsung Gear S3 Frontier

In generale, tutti questi dispositivi offrono un set di funzionalità di base più o meno comune a tutti e cioè permettono di gestire chiamate, notifiche, immagini, riproduzione musicale, activity & sleep tracker e il tutto senza togliere il proprio smartphone dalla tasca. Alcuni modelli più avanzati permettono di svolgere quanto appena detto anche in totale autonomia, senza dover essere necessariamente collegati allo smartphone. Chiaramente i modelli più performanti e/o con funzionalità aggiuntive e più precise sono giustificati da un costo maggiore.

Esistono poi i così detti *fitness-tracker* o *smartband*, simili agli smartwatch ma dalla forma più stretta che ricorda quella tipica dei bracciali e con funzionalità specifiche per lo sport. Questi dispositivi consentono di monitorare i movimenti, le calorie bruciate, la distanza percorsa e la qualità del sonno o direttamente sul display del bracciale oppure tramite lo smartphone.

1.2 Introduzione all'Health Monitoring

Finora è stata introdotta la tecnologia wearable da un punto di vista generale. Questa tesi, però, intende concentrarsi sul wearable applicato all'ambito dell'health monitoring, dunque è giunto il momento di trattare questa tematica.

Per health monitoring si intende un insieme di sistemi e risorse atti a monitorare lo stato di salute di un individuo. Fino a non molti anni fa, questo tipo di pratica veniva svolta soltanto nelle strutture sanitarie, come ad esempio gli ospedali o le cliniche private, ed era richiesto un personale esperto in grado di utilizzare correttamente determinati macchinari, oltre che capace di interpretare i dati clinici forniti in output. Inoltre, soltanto gli enti sanitari dispongono dei fondi e degli spazi fisici per acquistare e contenere tali apparecchiature. Infatti, da un lato il costo dei macchinari necessari per effettuare determinati esami è piuttosto elevato proprio perché destinato ad un uso professionale, dall'altro il volume fisico occupato da queste apparecchiature è piuttosto notevole, risultando ovviamente non adatte ad un uso domestico.

L'idea di concentrare la tecnologia necessaria per effettuare health monitoring in un piccolo e pratico dispositivo indossabile ha destato parecchio interesse negli ultimi anni. L'evoluzione tecnologica ha portato alla nascita di sensori sempre più piccoli e dal consumo quasi irrisorio; a questo si aggiungono anche le batterie che, sebbene le ridotte dimensioni, riescono a fornire energia sufficiente per alimentare il microcontrollore, i sensori e un modulo di comunicazione wireless.

Con un dispositivo smart di questo tipo un individuo può monitorare i propri parametri vitali autonomamente, senza dover necessariamente recarsi presso una struttura sanitaria. Naturalmente le prestazioni, l'efficienza e soprattutto l'affidabilità di tali dispositivi non sono

paragonabili a quelli utilizzati in ambito ospedaliero. Infatti, l'obiettivo non è in alcun modo quello di sostituire completamente questi ultimi, ma di realizzare dispositivi facilmente accessibili e configurabili da persone che non dispongono di competenze mediche.

Il progresso tecnologico ha portato alla nascita di sistemi per health monitoring più o meno complessi a seconda delle esigenze degli individui a cui sono destinati. Nei casi più semplici troviamo dispositivi indossabili destinati ad un pubblico di massa, basti pensare all'avvento degli smartwatch e delle smartband, che sono in grado di monitorare il battito cardiaco, l'attività fisica e la qualità del sonno. È opportuno precisare che questo genere di dispositivi è destinato o ad un uso sportivo non professionale o di puro appagamento personale e non possono sicuramente essere catalogati come dispositivi medici. L'idea alla base di uno smartwatch è quella di unire l'eleganza di un orologio con le funzionalità "smart" che esso può offrire, come il monitoraggio di una sessione di allenamento, notifiche di messaggi, chiamate in arrivo, app, sveglie, ecc. Nel caso in cui si soffra di determinate patologie, si sconsiglia di fare affidamento su questi dispositivi; al più possono fornire delle informazioni orientative, che andranno successivamente verificate con apparecchiature professionali.

I programmi di allenamento sportivo integrati negli smartwatch e nelle smartband si limitano a monitorare, in maniera approssimativa, lo sforzo fisico di un individuo durante una sessione di allenamento, oltre a registrare altri parametri di natura non medica come tempo, distanza percorsa, ecc. Normalmente la funzione principale di questi dispositivi è quella di registrare e monitorare il battito cardiaco, notificando all'utente quando si raggiungono valori troppo alti. Di recente sono stati sviluppati anche sensori in grado di rilevare il livello di ossigeno presente nel sangue, sebbene siano ancora un po' imprecisi.

Esistono inoltre dispositivi destinati espressamente a chi pratica sport come, ad esempio, le fasce da cardio che misurano la frequenza cardiaca in prossimità del cuore, dal momento che vengono indossate al torace. Questo genere di dispositivi è sicuramente più preciso di uno smartwatch o di una smartband e a differenza di questi ultimi vanno indossati soltanto durante la sessione di allenamento: nella vita quotidiana risulterebbero troppo invasivi.

Finora abbiamo trattato singoli dispositivi wearable che integrano diverse funzionalità e il cui scopo è offrire praticità, versatilità, comodità e in alcuni casi anche eleganza come dimostrato dai moderni smartwatch. Relativamente all'ambito dell'healthcare, su cui questa tesi intende focalizzarsi, è opportuno spendere alcune parole anche su una categoria di sistemi wearable non facilmente reperibili in commercio ma sicuramente protagonisti della ricerca odierna. Questi sistemi sono le body area network (BAN) e sono oggetto del seguente paragrafo.

1.2.1 Body area network (BAN)

Le body area network (BAN), a volte denominate anche body sensor network (BSN), sono costituite da un insieme di dispositivi wearable interconnessi tra loro e indossati contemporaneamente da un soggetto. In molti casi si parla anche di wireless body area network (WBAN) quando i dispositivi sono collegati tramite tecnologia wireless [7].

Abbiamo già parlato di come al giorno d'oggi c'è una forte tendenza a miniaturizzare i calcolatori in modo da essere sempre meno ingombranti e addirittura facilmente indossabili in qualsiasi parte del corpo. Tali dispositivi possono essere sia indossati esternamente al corpo, sia impiantati all'interno, ad esempio sottopelle. Quest'ultimo aspetto è particolarmente delicato, poiché impiantare piccoli dispositivi all'interno di

un organismo è chiaramente un'operazione rischiosa per la salute dello stesso. I vantaggi che si otterrebbero da una rete di sensori interamente impiantata nel proprio corpo sarebbero notevoli e sebbene la strada sia ancora lunga, la ricerca sta lavorando molto per rendere possibile tutto questo [8], che fino a pochi anni fa era pura fantascienza.

Una generica BAN è composta da BSU (body sensor units) e da un BCU (body central unit). I primi sono i sensori che si occupano di raccogliere i dati, mentre il secondo costituisce l'unità centrale che raccoglie ed elabora i dati. Come ormai comune nel mondo IoT, l'unità centrale può essere collegata ad un dispositivo smart (smartphone, tablet, ecc.) che fornisce un'interfaccia utente per configurare e gestire l'applicazione BAN, oltre a permetterne indirettamente l'accesso ad Internet.

Questa tipologia di reti è di particolare interesse in ambito di healthcare poiché permette di monitorare molteplici parametri clinici interagendo con diverse parti del corpo; a tal proposito si parla di medical body area network (MBAN). Applicazioni di questo tipo potrebbero risultare di grande aiuto per quelle persone che soffrono di patologie croniche come diabete, asma e problemi cardiaci. Inoltre, le BAN potrebbero essere utilizzate per eseguire specifici esami, per esempio le prove da sforzo, dove è necessario monitorare molteplici parametri per misurare il livello di stress fisico dell'intero organismo.

Infine, dal momento che possono essere indirettamente collegate ad Internet, sarebbe possibile usufruire di alcune funzionalità IoT come, ad esempio, la gestione real-time di una cartella clinica digitale tramite il quale i medici potrebbero monitorare da remoto lo stato di salute del proprio paziente. Inoltre, in caso di emergenza, il sistema potrebbe allertare le autorità di soccorso con un certo anticipo e quindi consentirebbe loro di poter intervenire tempestivamente. Questo aspetto non è banale, dal momento che moltissime volte anche un solo minuto di anticipo può salvare

la vita ad una persona. Tutti questi concetti sono alla base della cosiddetta telemedicina che permette ad un medico di effettuare diagnosi e cura di un paziente a distanza attraverso alcuni strumenti informatici.

1.2.2 Dispositivi wearable per Heath Monitoring in commercio

I dispositivi wearable con certificazione medica si differenziano per l'appunto perché forniscono misure precise e affidabili quando si deve monitorare lo stato di salute di un individuo, infatti sono progettati esclusivamente per questo scopo. Le funzionalità offerte da questi dispositivi possono essere le più varie a seconda della patologia di cui soffre il paziente e di quali parametri si desidera monitorare. Partiamo dal "classico" monitoraggio cardiaco (già ampiamente citato, ma inteso qui sotto un punto di vista più professionale) fino ad arrivare alle misurazioni di glucosio nel sangue per quelle persone affette da diabete e a dispositivi in grado di monitorare le convulsioni di soggetti che soffrono di epilessia.

Molti di questi dispositivi riproducono la tradizionale forma ad orologio, mentre altri necessitano di essere collocati in una specifica parte del corpo; ciò dipende dalla patologia del paziente e da quali parametri si desidera monitora. Ricordiamo, inoltre, che ormai viviamo in una società smart, in cui pressoché tutti i dispositivi sono direttamente o indirettamente connessi alla rete Internet e questa tendenza sta coinvolgendo anche il wearable in ambito medico.

Diamo adesso uno sguardo al mercato odierno, facendo una panoramica dei principali dispositivi wearable per health monitoring analizzandone le loro caratteristiche comuni e gli ambiti di interesse.

La maggior parte dei dispositivi wearable per health monitoring certificati a livello medico sono basati sul Bluetooth Low Energy (BLE); la scelta di questo protocollo permette di usufruire di diversi vantaggi. Prima

di tutto, poiché questi dispositivi sono gestiti e supportati da una app dedicata che gira su smartphone, si risolvono i problemi legati alla compatibilità, poiché qualsiasi smartphone supporta la connettività BLE. Inoltre, come suggerisce il nome, questa particolare versione del protocollo Bluetooth è ottimizzata per il risparmio energetico e si presta perfettamente per essere adoperata a bordo di dispositivi alimentati a batteria. Più avanti vedremo più in dettaglio le caratteristiche e i vantaggi di questo protocollo.

Come accennato pocanzi, questi dispositivi attualmente in commercio sono affiancati da una app dedicata che offre totale supporto ed estensione delle funzionalità. Infatti, tramite essa è possibile configurare le opzioni principali, sincronizzare i dati, elaborarli e infine visualizzarli sotto forma di grafici o semplici testi in modo da presentare le informazioni all'utente in modo chiaro e comprensibile. Il vantaggio sta nel fatto che tutte queste operazioni possono essere svolte in totale autonomia attraverso il proprio smartphone. Vediamo adesso singolarmente ognuno di questi dispositivi, analizzando le funzionalità che offrono e le loro performance. Come si potrà notare ognuno di essi si distingue per una propria funzionalità specifica relativa ad una determinata patologia o a determinati parametri clinici.

1.2.3 Nonin WristOx$_2$ Model 3150 with BLE

Il WristOx2 è un pulsiossimetro professionale dalle sembianze di un orologio digitale con supporto della connettività BLE al fine di interfacciarsi con la relativa app per smartphone. Questo dispositivo è ideale per quelle persone che soffrono di

Figura 1.6 - WristOx2

ipossia, ovvero una patologia che porta ad avere poca ventilazione polmonare: il livello di ossigeno presente nel sangue è un indicatore di tale patologia. Inoltre, può essere indicato per chi soffre di problemi cardiaci, ragion per cui necessita di monitorare quotidianamente le proprie pulsazioni.

Il dispositivo è in realtà composto da due componenti. La prima è quella che va indossata al polso e comprende un display dove vengono visualizzate le pulsazioni, il livello di ossigenazione del sangue e la carica residua, la seconda è un sensore esterno che va indossato all'estremità dell'indice e si collega tramite un cavetto alla componente da polso; questo sensore è fondamentale per misurare l'ossigenazione del sangue [9].

Analizzando le performance di questo dispositivo, stando a quanto dichiarato dal produttore [10], si può notare come esso offra un'elevata accuratezza e autonomia. Si riporta una precisione di +/-3 cifre per quanto concerne sia il valore di ossigenazione del sangue che delle pulsazioni se la misurazione viene effettuata in movimento, mentre l'errore scende a +/-2 cifre nel caso di misurazioni a riposo o in caso di bassa perfusione. Per quanto riguarda l'autonomia invece, a seconda della modalità di utilizzo, può variare drasticamente. I valori sono riportati nella seguente tabella tratta dalla documentazione ufficiale.

Power Requirements:	Two AAA (1.5V) batteries		
Battery Life (expected minimum): **NOTE:** Based on testing new and fully-charged batteries. See footnotes for brands used. Refer to battery manufacturers' operator's manuals for instructions for use.	**Alkaline AAA[a]**	**Rechargeable AAA (700 mAh)[b]**	**Rechargeable AAA (1050 mAh)[c]**
Storage: MVI Display Mode off: MVI Display Mode on:	9 months 25 days	Not specified	Not specified
Operating without Bluetooth Connection, continuous use:	53 hours	36 hours	52 hours
Operating with Bluetooth Connection, continuous use:	44 hours	24 hours	31 hours
			D)
	(2.0 in. x 2.9 in. x 0.75 in.)		

Figura 1.7 – Dettagli sull'autonomia del WristOx2

1.2.4 Embrace

Questo dispositivo tutto italiano è un bracciale smart capace di rilevare con un certo anticipo una crisi epilettica, inviando tempestivamente un allarme alla persona che assiste il soggetto. Embrace è il primo dispositivo wearable ad aver ricevuto il riconoscimento dalla FDA (Food and Drug Administration, cioè l'ente americano che si occupa della regolamentazione di alimenti e farmaci immessi sul mercato) come dispositivo medico in ambito neurologico [11].

Figura 1.8 - Embrace

Quello che fa è monitorare l'attività elettrica a livello cutaneo attraverso la sudorazione. Essa, infatti, riflette i cambiamenti nell'attività delle ghiandole sudoripare, che a sua volta riflettono alterazioni del sistema nervoso relativamente ad aspetti emotivi.

Embrace ha riportato risultati strabilianti al punto da riuscire a rilevare il 94% delle crisi epilettiche.

Questo dispositivo integra un sensore EDA (Electrodermal Activity), che serve a misurare le variazioni elettriche a livello cutaneo come accennato pocanzi, un sensore di temperatura, un accelerometro e un giroscopio per misurare i movimenti; questi sensori sono indispensabili per effettuare il monitoraggio del sonno. In questo caso, tale funzionalità si concentra sull'attività cerebrale, a differenza del semplice sleep tracker fornito da un comune smartwatch.

Analizzando l'aspetto più tecnico, si possono consultare le seguenti tabelle [12] che riportano le specifiche della batteria, dei sensori e del modulo Bluetooth a bordo di Embrace. In particolare, sono interessanti le frequenze di campionamento di ciascun sensore che indicano con quale cadenza vengono registrati i dati. Questi parametri influiscono non poco sull'autonomia del dispositivo, che in questo caso viene dichiarata superiore alle 48 ore.

Sensors

EDA Sensor: Sampling frequency: 4 Hz (fixed). • Resolution: 55 fSiemens. • Range: 0 µSiemens – 80 µ– 80 e: • AC excitation, 4Hz, max. • Electrodes (applied parts): Placement on either the ventral (inner) wrist or dorsal (outer) wrist.	**3-Axis accelerometer** • Sampling frequency: 32 Hz • High sensitivity motion detection on 3 axes: • Range ± 16g. • Resolution: 16 bits
Temperature Sensor • Sampling frequency: 1 Hz • Range: -20°C to 70°C • Resolution: 0.15°C	**Gyroscope** • Sampling frequency: 32 Hz • High sensitivity motion detection on 3 axes • Range ± 500 degrees/sec. • Resolution: 16 bits

Power

Battery Characteristics 3.7V Li-ion 90 mAh minimum capacity Battery life 2 yr Charge cycles 600	**AC Adapter Specifications** Input Voltage: 5 V Output Current: 100 mA USB A (charge only, no data transfer)

Bluetooth

Operation frequency: 2402-2480 MHz Equipment class DTS Max radiated power: 87,90 dBµV/m (at 3 m distance) Modulation GFSK	Channel spacing: 2 MHz Channel bandwidth: 2 MHz Number of channels: 40

1.2.5 Heart Guide by Omron

Il mercato degli smartwatch come già accennato è in continua crescita e quasi tutte le aziende hanno introdotto un proprio modello. La medesima cosa ha fatto Omron, azienda leader in ambito di healthcare, famosa per la produzione di diverse apparecchiature in ambito sanitario come, ad esempio, i misuratori di pressione sanguigna. A tal proposito l'azienda ha sfruttato la sua esperienza realizzando il primo smartwatch in grado di misurare, oltre le pulsazioni, anche la pressione sanguigna [13].

Figura 1.9 - HeartGuide by Omron

Questo smartwatch si distingue dai suoi simili perché non adopera i comuni sensori a led che misurano la quantità di luce dispersa dal flusso sanguigno, bensì utilizza lo stesso sistema utilizzato dai misuratori di pressione professionali. Infatti, la caratteristica vincente di questo dispositivo è la camera d'aria gonfiabile nascosta nel lato interno del cinturino che consente di effettuare delle vere e proprie misurazioni della pressione arteriosa in modo molto preciso.

1.2.6 Wearable biosensor by Philips

Un approccio simile al precedente è stato adottato da Philips che ha sviluppato un biosensore [14] dalla forma di una patch adesiva che va applicata sul petto all'altezza del cuore. Questo dispositivo racchiude un accelerometro MEMS a tre assi, un sensore di temperatura e degli elettrodi ECG. Questi sensori permettono di monitorare diversi parametri ed eventi: frequenza cardiaca, frequenza respiratoria, temperatura della pelle, postura del corpo, rilevamento delle cadute e conta passi.

Figura 1.10 - Wearable biosensor by Philips

A differenza di altri dispositivi wearable che sono orientati ad un più privato consentendone la gestione tramite smartphone, questo biosensore nasce per essere utilizzato in ambito ospedaliero, al fine di semplificare e migliorare l'assistenza verso i pazienti. Il dispositivo invia automaticamente i dati ad un sistema centrale che gestisce gli eventuali allarmi e avvisa tempestivamente i medici.

1.2.7 Dexcom G6

Nettamente differente ma di notevole importanza è il Dexcom G6 [15], un dispositivo alle persone affette da diabete. A differenza dei precedenti dispositivi trattati che prevedevano di essere indossati al polso, questo va applicato sotto la cute attraverso un apposito strumento che viene fornito in dotazione. Esso consiste in un piccolo sensore che effettua periodicamente le misurazioni di glucosio e invia i dati ad un dispositivo di visualizzazione che può essere quello fornito dall'azienda o un comune smartphone. L'app ovviamente è fondamentale al fine di gestisce gli eventuali allarmi in caso di glicemia alta o bassa, riuscendo addirittura a prevedere quei bruschi cali tipici di questa patologia. Inoltre, memorizza tutto lo storico dei dati raccolti, consentendo al soggetto di monitorare l'andamento giornaliero del proprio valore glicemico.

Figura 1.11 - Dexcom G6

2 Applicazioni distribuite e comunicazione

Nel capitolo precedente sono state illustrate le più comuni applicazioni wearable sia di uso comune che specifiche per health monitoring. È facile notare come tutte queste applicazioni seguano uno schema comune composto essenzialmente da tre componenti: il dispositivo wearable, uno smartphone e il server-cloud al quale l'app si connette. La completa e corretta esecuzione dell'intera applicazione prevede infatti che il dispositivo wearable si interfacci con lo smartphone tramite un'apposita app, quasi sempre proprietaria, con il quale sincronizzerà tutti i dati raccolti e dal quale riceverà diversi tipi di comandi. L'app, a sua volta, elaborerà i dati ricevuti e ne invierà altri ad un server-cloud che li memorizzerà associandoli all'account dell'utente. In questo modo i dati raccolti potranno essere condivisi con altri utenti oltre che utilizzati dalla stessa azienda per effettuare misure statistiche.

Come già discusso nell'introduzione del capitolo precedente, in questa tesi si vuole presentare l'idea di un'applicazione distribuita in ambito di smart environment in cui dispositivi fissi e wearable opportunamente interconnessi condividono i dati raccolti con un hub centrale. L'obiettivo è quindi l'integrazione dei dispositivi wearable con gli smart environment orientati, nell'analisi in questione, all'health monitoring. In uno scenario come quello appena proposto, lo schema adottato dai dispositivi visti nel precedente capitolo risulterebbe restrittivo e soprattutto incompleto poiché il singolo dispositivo wearable sarebbe vincolato a comunicare esclusivamente con uno smartphone. Inoltre, se il soggetto da assistere è una persona anziana che, come accade spesso, non ha molta dimestichezza

con la tecnologia odierna, il modello basato esclusivamente su smartphone risulterebbe impraticabile.

A seguire, vedremo una panoramica sugli smart environment e le applicazioni distribuite, infine saranno trattati sommariamente i principali protocolli di comunicazione utilizzati in questo ambito. Nel capitolo successivo, invece, vedremo più in dettaglio i vari aspetti del sistema proposto insieme allo sviluppo di un'applicazione wearable per health monitoring.

2.1 Smart environment

Al giorno d'oggi la parola "smart" e il concetto che essa racchiude è sempre più frequente nella nostra società. Assistiamo ogni giorno alla nascita di nuovi dispositivi smart in grado di assisterci durante la vita di tutti i giorni con l'obiettivo di semplificare, migliorare e automatizzare le attività che ognuno di noi svolge quotidianamente.

Alcuni di questi dispositivi, come già visto nel capitolo precedente, sono progettati per operare individualmente, mentre altri sono destinati a costituire collettivamente una rete di nodi che dà vita a ciò che viene definito smart environment. I nodi possono essere equipaggiati di sensori per il monitoraggio di una specifica grandezza fisica ambientale, come ad esempio la temperatura e l'umidità, oppure una certa attività, come la presenza di persone nell'ambiente. Essi possono anche disporre di attuatori che si mettono in funzione non appena il sistema rileva una certa anomalia nell'ambiente attraverso i sensori.

Per esempio, potremmo immaginare un sistema domotico capace di monitorare, attraverso specifici nodi sensori, la qualità dell'aria ed in particolare la presenza di fumo e gas. Nel caso venga rilevata una situazione di pericolo, il sistema potrebbe attivare degli appositi attuatori che aprono

le finestre della stanza per favorire il ricircolo dell'aria ed inoltre potrebbe anche segnalare l'allarme ai soccorsi.

In uno smart environment ogni nodo coopera con gli altri scambiando dati che vengono progressivamente elaborati fino ad estrarre informazioni di alto livello circa lo status dell'ambiente o la criticità di eventuali situazioni anomale. Inoltre, uno smart environment è capace di apprendere "osservando" l'ambiente e i comportamenti delle persone coinvolte. In questo modo il sistema tende ad essere il più autonomo possibile, limitando al minimo le interazioni con l'utente.

Una infrastruttura di questo tipo è applicabile ai luoghi e ai contesti più vari; quelli principali sono sicuramente le case, le città e le industrie, da cui derivano i termini smart home, smart cities e smart manufacturing.

Le smart home stanno diventando sempre più popolari negli ultimi anni, soprattutto dopo l'avvento di sistemi intelligenti come Amazon Alexa e Google Home. Questi dispositivi possono essere connessi ad una moltitudine di altri dispositivi smart collocati all'interno dell'abitazione, il tutto appoggiandosi alla rete WiFi domestica.

Questi dispositivi sono dotati di assistenti vocali con il quale l'utente può interagire ponendo domande o impartendo specifici comandi. È infatti possibile chiedere in tempo reale e interattivamente informazioni meteo, notizie, informazioni sul traffico, e molto altro, il tutto semplicemente attraverso un dialogo orale. Altra possibilità è quella di impartire comandi come accendere/spegnere le luci o gli elettrodomestici, impostare sveglie, riprodurre musica e molto altro. Tutto ciò è possibile grazie alla moltitudine di dispositivi smart compatibili con Amazon Alexa e Google Home che le stesse aziende mettono a disposizione nei propri store. Tra questi dispositivi smart troviamo lampadine, prese elettriche, interruttori, termostati, videocamere di sorveglianza, altoparlanti, ecc.; la scelta è sicuramente molto vasta.

La crescente popolarità di questi dispositivi è dovuta all'idea che ci sta dietro, ovvero quella di poter comporre autonomamente la propria smart home acquistando i singoli componenti, la cui installazione è semplice e immediata.

Il concetto di smart environment, come abbiamo detto, è anche applicabile alle città. Una smart city è definita tale quando le tecnologie di comunicazione impiegate all'interno della città, migliorano le infrastrutture e la conseguente fruizione dei servizi da parte dei cittadini, il tutto in uno scenario ecosostenibile. In un progetto di questo tipo sono coinvolti molti settori, ad esempio quello dei trasporti pubblici, dell'energia, della sicurezza urbana, del monitoraggio ambientale e dei sistemi di comunicazione delle informazioni.

Gli smart environment non si limitano ad abbracciare soltanto abitazioni e città, ma si estendono anche alle industrie. La smart manufactoring mira ad ottimizzare il lavoro all'interno delle industrie, sia da un punto di vista produttivo che di sicurezza del personale. Quindi da un lato il sistema cerca di semplificare il lavoro del personale e di massimizzare la produzione, dall'altro invece garantisce, attraverso un apposito sistema di monitoraggio continuo, la sicurezza dello stesso personale di lavoro. A tal proposito possiamo immaginare una fabbrica in cui migliaia di sensori monitorano ogni tipo di attività critica che si svolge all'interno dell'ambiente di lavoro e in caso di pericolo il sistema non solo segnala immediatamente la criticità ma avvia anche un protocollo di sicurezza per correggere tali anomalie per prevenire situazioni drammatiche.

È interessante notare come in ambito industriale, la sicurezza dell'ambiente rivesta un aspetto maggiormente significativo rispetto agli ambiti domestici e urbani, ciò è chiaramente legato alla maggiore presenza di rischi che gli individui corrono all'interno di una fabbrica; ovviamente tutto è sempre relativo al tipo di industria.

Dopo questa introduzione agli smart environment è giunto il momento di trattare le applicazioni che stanno alla base di questi ambienti intelligenti: si tratta di applicazioni distribuite.

2.2 Applicazioni distribuite

A differenza di un'applicazione tradizione che viene eseguita su un singolo calcolatore, un'applicazione distribuita, come dice il nome, è composta da una serie di processi distribuiti, per l'appunto, su più calcolatori interconnessi tra loro. Ogni dispositivo costituisce un nodo della rete e può comunicare con gli altri nodi o direttamente o attraverso un dispositivo centrale. Quest'ultimo è spesso il punto di accesso principale da parte dell'utente, il quale, tramite un'apposita interfaccia (app su smartphone, dispositivo o software dedicato, ecc.) può accedere all'intera infrastruttura concependola come se fosse un'unica applicazione. A tal proposito, una funzionalità non trascurabile spesso presente nelle applicazioni distribuite è la possibilità di accedere da remoto oltre che dalla rete locale.

Un'architettura di questo tipo è fortemente basata sul modello client/server, in cui i nodi client contattano i nodi server i quali rispondono fornendo servizi di vario genere [16]. Ovviamente alcuni nodi possono anche essere ibridi, cioè rivestire sia il ruolo di client che di server. Relativamente a quanto appena detto, è opportuno sottolineare come ogni nodo rivesta un'importanza differente, ciò significa che in alcuni casi lo spegnimento o il malfunzionamento di un singolo nodo potrebbe anche compromettere il funzionamento dell'intera rete; di solito i nodi più critici sono quelli server. Inoltre, un aspetto non indifferente da considerare in una infrastruttura di questo tipo è la mancanza di una sincronizzazione comune dovuta alla moltitudine eterogenea di dispositivi coinvolti, ognuno

operante ad una differente frequenza di clock. Ciò significa che non è possibile gestire la concorrenza dei vari processi come in una applicazione tradizionale.

Alla base dello sviluppo di un'applicazione distribuita c'è il problema di interconnettere dispositivi che fanno uso di tecnologie di comunicazione hardware e software differenti. Per ovviare a questo problema, ogni applicazione distribuita fa uso dei middleware, cioè un insieme di strumenti di sviluppo che rendono possibile la comunicazione all'interno di una rete dove si utilizzano una certa varietà di protocolli differenti. Tali strumenti possono essere DBMS, Web server, Applicazion server, XML, SOAP, ecc., e, come sappiamo, permettono all'utente di ottenere un ottimo livello di astrazione in fase di sviluppo.

Per fare un esempio, la stessa rete Internet può essere considerata una grande applicazione distribuita, dove miliardi di risorse sono distribuite in tutto il mondo ma sono accessibili da qualsiasi host collegato alla rete e dove la comunicazione tra i nodi fa uso di una grandissima varietà di protocolli.

Le applicazioni distribuite di interesse ai fini di questa tesi sono quelle relative agli smart environment. Una generica smart home composta dai dispositivi offerti da Amazon e Google implementa a tutti gli effetti un'applicazione distribuita che rispetta tutte le caratteristiche sopra elencate. Ogni dispositivo implementa un software specifico per la sua piattaforma hardware sottostante ed è in grado, attraverso un middleware, di comunicare con l'hub centrale.

Come già detto, i dispositivi di questi due marchi importanti utilizzano la rete WiFi domestica per interconnettersi tra loro. La scelta di questo protocollo non è casuale, dal momento che ormai in quasi tutte le abitazioni è presente un router, dunque, risulta semplice ed immediato appoggiarsi ad una rete già esistente e di facile configurazione.

In realtà esistono molteplici protocolli di comunicazione che si prestano bene per interconnettere i dispositivi di una smart home e nel prossimo paragrafo saranno illustrati quelli principali, tra cui anche il WiFi sopra citato.

2.3 Protocolli di comunicazione

Prima di parlare singolarmente dei protocolli di comunicazione wireless è opportuno fare un confronto con i protocolli wired.

Il vantaggio principale della comunicazione wireless è sicuramente la mobilità, che risulta indispensabile in ambito wearable affinché l'individuo possa muoversi liberamente senza essere fisicamente collegato ad un altro dispositivo. I dispositivi fissi, invece, data la loro staticità, possono usufruire sia di collegamenti wired che wireless, anche se ultimamente i produttori stanno propendendo più verso quest'ultimo tipo, dal momento che ciò semplifica moltissimo l'installazione degli stessi, che non richiede il passaggio di alcun cavo all'interno dell'abitazione.

La comunicazione wireless, però, comporta anche alcuni svantaggi che bisogna sempre tenere in considerazione. A differenza delle reti wired dove i dati sono vincolati a transitare attraverso un cavo, nelle reti wireless, per l'appunto, i dati transitano attraverso onde elettromagnetiche che si propagano liberamente nello spazio. Ciò comporta un maggiore rischio di sniffing dei dati, ovvero un dispositivo terzo potrebbe captare la comunicazione e leggere, modificare e reinserire a proprio piacimento i pacchetti di dati.

Il secondo svantaggio delle reti wireless consiste nelle interferenze con altre comunicazioni che adoperano la stessa frequenza, ragion per cui per ogni protocollo viene standardizzata una frequenza ben precisa. Inoltre, il segnale potrebbe anche essere ostacolato da oggetti fisici che ne limitano la

copertura: in ambito domestico le pareti che delimitano le stanze costituiscono un ostacolo rilevante. Ovviamente ogni tipo di segnale e protocollo risulta essere più o meno robusto a questo fattore.

Infine, ma non meno importante, bisogna ricordare che la velocità di trasmissione nelle reti wireless rispetto alle reti wired è ovviamente minore e si riduce con l'aumentare della distanza tra i due dispositivi.

In contesti domestici, come accennato pocanzi, ciò che bisogna principalmente considerare è la copertura del segnale e le eventuali interferenze con altri segnali. Per quanto concerne la sicurezza, è sicuramente importante adottare le giuste contromisure, che non necessitano di essere le stesse adottate in ambienti affollati. In altri termini in una smart home non è necessario adottare le stesse tecniche di crittografia utilizzate nei progetti di smart cities. Inoltre, è bene ricordare le esigue risorse di un dispositivo wearable che potrebbe non supportare le più complesse tecniche di crittografia moderna.

Vediamo adesso una breve panoramica dei principali protocolli di comunicazione wireless, cercando di capire quali si prestano meglio in ambito wearable.

2.3.1 WiFi

Il protocollo Wi-Fi è famoso per essere lo standard wireless di tutti i router/switch utilizzati in ambito domestico e aziendale. Tale protocollo è integrato in qualsiasi dispositivo portatile come notebook, smartphone, smartwatch, tablet, ecc. e permette la connessione ad un router che a sua volta permetterà la navigazione in Internet. Il WiFi offre notevoli vantaggi sotto molti aspetti, garantendo una buona copertura del segnale e un'elevata velocità di trasmissione. Inoltre, i moderni standard WiFi hanno reso questo protocollo molto robusto dal punto di vista della sicurezza.

Tutti questi aspetti positivi si pagano però con un elevato consumo energetico. Quest'ultima è la ragione per cui il WiFi non si presta bene ad applicazioni in ambito wearable, dove questo aspetto è fondamentale data l'esigua capacità della batteria a bordo del dispositivo.

2.3.2 Z-Wave

Un protocollo che invece offre una soluzione a basso consumo energetico è Z-Wave [17]. A differenza del WiFi dove ogni dispositivo si collega individualmente ad un'unica unità centrale (il router), nel protocollo Z-Wave ogni dispositivo (slave) costituisce un nodo di una rete mesh a cui fa capo un controller (master). Ogni nodo può svolgere funzioni di inoltro comunicando con gli altri nodi della rete fino ad arrivare al controller. Questa è la caratteristica che contraddistingue Z-Wave dagli altri protocolli di comunicazione e lo rende ideale in ambienti domotici dove si hanno diversi dispositivi da interconnettere.

Sfortunatamente i moduli Z-Wave sono piuttosto costosi e per tale ragione non vengono praticamente adoperati in ambito wearable. Tuttavia, questo protocollo, a livello tecnico, rispetta tutti i requisiti del wearable e nulla vieta di utilizzarlo in questo ambito.

2.3.3 ZigBee

Un'ottima alternativa che costituisce un po' una via di mezzo tra WiFi e Z-Wave è ZigBee [18]. Infatti, questo protocollo colma, per certi versi, alcune lacune dei due protocolli precedenti e ne eredita gli aspetti positivi. Dal punto di vista energetico, a differenza del WiFi, ZigBee consuma davvero pochissimo, a fronte però di una ridotta velocità di trasmissione di circa 125 kbps. Invece, dal punto di vista economico, a differenza dei moduli

Z-Wave, il costo dei moduli ZigBee si aggira nell'ordine di pochissimi dollari.

Dal punto di vista macrostrutturale l'obiettivo di ZigBee, come per Z-Wave, è di definire una rete mesh, ma la microstruttura è molto simile a quella prevista dal WiFi. Una rete ZigBee è di solito composta da tre moduli: il Coordinator (ZC), i Router (ZR) e gli End Devices (ZED). Il primo è il dispositivo che inizializza la rete mentre i secondi gestiscono le comunicazioni tra gli end devices, realizzando così una serie di sottoreti che possono comunicare tra loro attraverso i router. L'idea alla base è quella di creare una rete di comunicazione su misura per ambienti domotici ma anche industriali, mantenendo al minimo il consumo energetico. Per tutte le ragioni sopra esposte, sicuramente ZigBee risulta essere un ottimo candidato per essere utilizzato in ambito wearable.

2.3.4 Bluetooth

Come si è già potuto immaginare dall'introduzione nel capitolo precedente, il protocollo Bluetooth è quello maggiormente utilizzato in ambito wearable. Al di là dei vantaggi tecnici offerti da questo protocollo, la scelta di utilizzare questo protocollo risiede nel fatto che già dispositivi come notebook, smartphone, tablet e tutti i sistemi operativi a bordo di essi, supportano nativamente il Bluetooth e dunque qualsiasi problema legato alla compatibilità viene del tutto superato. Inoltre, in questo modo si ha una vasta gamma di dispositivi con cui ci si può interfacciare.

Nel 2011 è stata presentata una variante del protocollo Bluetooth orientata al basso consumo, che prende appunto il nome di Bluetooth Low Energy, colloquialmente abbreviato BLE. Questa nuova versione è stata sviluppata appositamente per dispositivi alimentati a batteria con esigue risorse energetiche come gli stessi smartphone ma soprattutto dispositivi

per la domotica, per la salute, per il fitness e qualsiasi genere di dispositivo wearable. Inoltre, le dimensioni di un modulo BLE sono molto contenute, condizione ideale per essere installato su piccoli dispositivi come, ad esempio, smartwatch e smartband.

Il BLE [19] è stato introdotto nelle specifiche 4.0 del Bluetooth, che non definiscono una retrocompatibilità con il precedente Bluetooth Base Rate/Enhanced Data Rate (BR/EDR), ovvero il Bluetooth "classico", ma consentono ai dispositivi di implementare o solo il sistema BLE o entrambi. Per esempio, uno smartphone moderno implementa ovviamente entrambi i sistemi. Dal punto di vista hardware i dispositivi "dual-mode" possono godere del vantaggio di condividere la stessa antenna radio, poiché entrambi i sistemi operano alla stessa frequenza radio di 2.4GHz. La possibilità del dual-mode garantisce una piena compatibilità con la moltitudine di dispositivi in circolazione.

Al giorno d'oggi, il BLE è stato integrato anche in diverse schede di sviluppo, come ad esempio il RaspberryPi 3 che integra un modulo unico BLE e WiFi. Dunque, data la sua immensa popolarità fra i dispositivi presenti nel mercato, che siano essi prodotti destinati al mercato di massa o schede di sviluppo, il BLE è stato scelto come protocollo di comunicazione per questo progetto. Nel prossimo capitolo sarà illustrato in maniera più approfondita il suo funzionamento.

2.3.5 Altri protocolli

Per concludere questa panoramica sui protocolli di comunicazione, sebbene non saranno utilizzati in questo progetto, citiamo per completezza altri due protocolli poco diffusi ma orientati al basso consumo: EnOcean e LoRa. Il primo permette trasmissioni fino a 30 metri all'interno degli edifici, e fino a 300 metri all'esterno. LoRa invece è una soluzione a lungo raggio,

che permette trasmissioni fino a 30 Km in zone rurali, e fino a 2 Km in contesti urbani, per cui si presta bene per applicazioni come Smart Cities, ma risulta sovradimensionato per applicazioni indoor come nel caso di questo progetto.

2.4 Gestione dei dati in cloud

Un aspetto importante che coinvolge quasi tutte le applicazioni distribuite è la gestione dei dati attraverso il cloud. Questa caratteristica coinvolge anche la maggior parte dei dispositivi wearable come quelli visti nel precedente capitolo. Infatti, attraverso la app su smartphone, si collegano a determinati server nei quali caricano i dati raccolti associandoli al profilo dell'utente.

Un tempo spettava al singolo dispositivo la gestione e la memorizzazione dei dati raccolti, poiché esso lavorava in maniera del tutto autonoma e non era collegato ad alcuna rete. Come esempio, possiamo citare le bilance pesa persone: quelle più tecnologiche avevano la possibilità di memorizzare i dati e fare alcune stime circa la salute del soggetto attraverso la lettura del peso corporeo. Oggi, nell'era IoT, esistono anche le bilance smart che, oltre a fornire particolari funzionalità extra rispetto ad una comune bilancia, possono essere collegate tramite app allo smartphone e quindi, di conseguenza, ad Internet. Ciò permette di usufruire di tutti i servizi che il mondo di Internet offre come la possibilità di condividere i propri dati con altri utenti in una maniera del tutto social, ma anche la memorizzazione dei dati nei cloud server proprietari. In questo modo se, ad esempio, l'utente dovesse cambiare smartphone per un qualsiasi motivo, non dovrà riconfigurare il dispositivo da zero ma dovrà semplicemente effettuare il login dalla app e troverà disponibile tutto lo storico dei propri dati.

Tutto ciò chiaramente stravolge il concetto di una semplice bilancia pesa persone, la cui funzione di base si riduce ad una piccola parte dell'intera applicazione. Con l'evolversi del mondo IoT, questa "rivoluzione" ha coinvolto non solo le bilance ma anche molti altri dispositivi con conseguente aumento dei dati da memorizzare associandoli però ad uno stesso profilo utente. Grazie al cloud tutto ciò viene reso possibile senza alcuna difficoltà da parte dell'utente.

Se da un lato la tecnologia cloud offre notevoli funzionalità e migliora notevolmente la gestione dei dati da parte dell'utente, dall'altra espone gli stessi ai rischi presenti nel mondo di Internet. Ovviamente ogni azienda che offre questo tipo di servizio garantisce un ottimo livello di sicurezza per quanto riguarda la memorizzazione dei dati sui propri server; ciò si traduce nell'utilizzo di avanzati sistemi di crittografia per ridurre al minimo la probabilità che i dati vengano violati.

Inoltre, non dimentichiamo il tema della privacy, che spesso è al centro di molte polemiche quando si parla della gestione dei dati in-cloud. Ogni qual volta un utente si registra su una piattaforma, gli viene chiesto di accettare quanto riportato in un documento contenente tutte le politiche riguardanti il trattamento dei propri dati ai fini della salvaguardia della privacy. Purtroppo, nella maggior parte dei casi, questo documento viene accettato senza neppure essere letto e d'altra parte se l'utente non accettasse non potrebbe beneficiare del servizio.

Solitamente i dati raccolti dalle aziende vengono utilizzati per scopi statistici e per fare analisi di mercato ma talvolta questi dati vengono condivisi anche con aziende terze con il quale hanno instaurato una collaborazione. Un esempio che può far chiarezza su tutto questo argomento sono le pubblicità che ritroviamo su diversi siti dove ci vengono proposti prodotti simili a quelli che abbiamo precedentemente cercato o acquistato su qualche sito di e-commerce.

Sebbene i moderni servizi di cloud risultino abbastanza robusti agli attacchi informatici, non sono del tutto inviolabili. Purtroppo, ancora oggi capita non di rado che diversi account relativi a marchi importanti come Google, Apple e Amazon vengano violati, con conseguenze piuttosto spiacevoli, come la diretta divulgazione delle informazioni sottratte o, nei casi peggiori, anche minacce e riscatti in cambio dei propri dati. Questi fenomeni purtroppo non sono così rari e possono colpire sia personaggi famosi, i quali sono sicuramente più a rischio per via della loro posizione di spicco nella società, sia la gente comune.

In alcuni casi, quindi, sarebbe giusto offrire all'utente la possibilità di scegliere se condividere o meno i propri dati con i server senza limitare l'utilizzo dell'applicazione. Un piano di questo tipo sarebbe molto utile quando l'applicazione tratta dati particolarmente sensibili, ad esempio quelli riguardanti la propria salute, come nel caso delle applicazioni distribuite per health monitoring. Chiariamo subito che qualsiasi dato privato costituisce un'informazione sensibile e necessita di transitare in assoluta sicurezza attraverso Internet, ma è anche vero che non tutti i dati di questo tipo hanno lo stesso peso. Ad esempio, è ovvio che i dati anagrafici hanno un peso minore rispetto ad una serie di dati clinici riguardanti, ad esempio, una patologia neurologica. Seguendo questa idea, la gestione dei dati viene rimandata del tutto all'utente, che può decidere in assoluta autonomia se condividere i dati e soprattutto con chi.

Come vedremo a breve nel prossimo capitolo, il sistema proposto in questa tesi, dal momento che è orientato all'health monitoring e quindi tratta dati molto sensibili, adotterà la filosofia sopra descritta.

3 Sviluppo di un'applicazione per health monitoring

Nei capitoli precedenti abbiamo effettuato una vasta panoramica sul mondo dei dispositivi wearable e degli smart environment e abbiamo visto quale sia la tendenza delle principali aziende che operano in questo campo.

In questa tesi si vuole proporre l'idea di un sistema ibrido di smart environment in cui sia dispositivi wearable che fissi, opportunamente coordinati da un hub centrale, cooperano per monitorare e assistere un soggetto all'interno della propria abitazione. Al giorno d'oggi, con l'aumentare dell'età media della popolazione, diventa sempre più necessaria la realizzazione di sistemi di home assistant e health monitoring al fine di salvaguardare la salute dei pazienti all'interno della propria abitazione. A tal proposito, in ambito di ricerca troviamo modelli di smart environment focalizzati al riconoscimento delle attività e dei contesti quotidiani all'interno di un'abitazione [20]. L'idea di base è quella di far apprendere al sistema i vari contesti che si possono verificare in un ambiente domestico, al fine di distinguere le situazioni ordinarie da quelle anomale.

In uno smart environment orientato all'health monitoring, i dispositivi fissi si occupano di monitorare tutte quelle grandezze fisiche e le attività esterne all'individuo. Queste possono essere parametri ambientali che definiscono la qualità dell'aria (temperatura, umidità, presenza di fumo e gas, ecc.) oppure la presenza e l'eventuale posizione dell'individuo all'interno dell'abitazione. D'altra parte, i dispositivi wearable si occuperebbero di monitorare tutti quei parametri che è possibile acquisire soltanto a diretto contatto con il soggetto.

Inoltre, non si esclude la presenza di dispositivi medici in versione smart come, ad esempio, un misuratore di pressione sanguigna, un pulsiossimetro oppure un glucometro, che potrebbero essere connessi alla rete per condividere i dati clinici raccolti. Tutte queste informazioni verrebbero raccolte ed elaborate da un'unità centrale che costituirà anche il punto di accesso per l'utente tramite un'apposita interfaccia.

Oltre ai dati numerici, i dispositivi possono anche inviare veri e propri eventi. Alcuni dispositivi sia fissi che wearable potrebbero implementare una specifica applicazione in grado di trasformare i dati in eventi ed inviarli all'unità centrale che dovrà soltanto discriminare l'evento, al fine di eseguire la routine corrispondente. Un approccio del genere potrebbe essere semplificato se i nodi si scambiassero frammenti di codice eseguibile, seguendo il modello delineato da [21] e [22]. In questo scenario i nodi invierebbero una semplice stringa (word) all'unità centrale, la quale, con l'ausilio di un interprete, eseguirà il codice ad essa associato. Questo modello semplificherebbe la comunicazione, dal momento che non esisterebbero più messaggi di richieste e di risposte e gli eventi si ridurrebbero a semplici word. Inoltre, con un interprete interattivo a bordo di ogni nodo sarebbe possibile la programmazione diretta senza necessità di ricompilare il sorgente e ricaricare il binario dell'applicazione sul dispositivo [23]. Quanto appena detto darebbe notevoli benefici in termini di prestazioni e semplificherebbe lo sviluppo e le modifiche in fase di test.

La fase sperimentale di questo progetto si concentrerà però sullo sviluppo di un'applicazione per health monitoring in grado di rilevare un evento di allarme e di comunicarlo all'unità centrale. Un sistema di esempio potrebbe comprendere un dispositivo wearable che implementa un'applicazione per il rilevamento delle cadute. Nel caso in cui questo evento si verifichi, esso lo comunicherebbe all'istante all'unità centrale che provvederà ad eseguire i dovuti accertamenti grazie all'ausilio di alcuni

dispositivi fissi, ad esempio delle videocamere, e in caso positivo segnalerà tempestivamente l'allarme a chi di competenza.

Come tutti gli smart environment anche questo sistema potrebbe essere connesso ad Internet per beneficiare di tutti i relativi servizi, come ad esempio il cloud. Come accennato alla fine del precedente capitolo, però, si è scelto di non adottare questo schema architetturale, dal momento che le informazioni trattate dal sistema sono per loro natura particolarmente sensibili. Ad esempio, se il sistema prevedesse un impianto di videosorveglianza, tramite il quale una persona può controllare da remoto un proprio parente infermo, è ragionevole comprendere che i filmati vogliano essere memorizzati soltanto in locale e non caricati su un cloud remoto.

Come da titolo, questa tesi intende concentrarsi sull'aspetto wearable del sistema proposto e in particolare si è scelto di affrontare il problema del rilevamento delle cadute, già citato pocanzi in un esempio. Seguirà un'analisi delle principali problematiche relative a questo tema molto delicato e successivamente vedremo lo sviluppo di un'applicazione wearable proponendo uno specifico algoritmo e valutandone l'efficienza. Nel capitolo successivo vedremo invece tutta la parte relativa al sistema di verifica con l'ausilio del dispositivo fisso; a tal proposito sarà presentato un modello di testing di generiche applicazioni distribuite attraverso la virtualizzazione di un sensore.

3.1 Rilevamento di cadute

Il problema del rilevamento delle cadute coinvolge, purtroppo, diversi soggetti, i quali, o per patologie o per senilità, soffrono di difficoltà motorie che causano loro instabilità nei movimenti, con conseguente rischio di cadere. Alcuni di questi soggetti sfortunatamente si ritrovano a vivere soli

in casa, per cui in caso di caduta non si ritroverebbero nessuno che possa assisterli.

Il rapporto mondiale dell'OMS sulla prevenzione delle cadute nell'anziano [24], ci dice che, ogni anno, circa il 28-35% delle persone di oltre 65 anni è soggetto ad una caduta, percentuale che sale a 32-42% negli ultrasettantenni. La maggior parte di queste cadute, circa il 60%, avvengono in casa. Ulteriori dati ci dicono che le cadute portano ad un 20-30% di infortuni medio-gravi e sono la causa principale del 10-15% di tutti gli accessi al pronto soccorso e di oltre il 50% dei ricoveri per infortunio di pazienti ultrasessantacinquenni. Inoltre, con l'aumentare dell'età e della fragilità, le persone anziane che subiscono un infortunio da caduta rischiano, purtroppo, di non recuperare completamente la precedente mobilità. Si pensi che il 20% delle persone che cadendo subisce una frattura al femore muore entro un anno dall'infortunio.

È evidente che questi dati siano molto allarmanti e che è necessario adottare delle valide misure preventive per evitare situazioni molto spiacevoli. Il sistema proposto in questa tesi si occupa proprio di questo, e cioè di rilevare una generica caduta e di comunicare tempestivamente l'allarme a chi di dovere, ad esempio un parente del paziente. Il sistema è costituito da un dispositivo wearable che si occupa di rilevare le eventuali cadute e da un'unità centrale che provvederà in maniera del tutto autonoma a comunicare l'allarme. In particolare, questa tesi approfondirà la parte wearable, focalizzando il lavoro sullo sviluppo di un'applicazione per il rilevamento delle cadute e sulla fase di testing supportata dal dispositivo fisso.

Purtroppo, in commercio non esistono ancora dispositivi particolarmente affermati in grado di offrire questa funzionalità. In ambito di ricerca, però, molti si sono attivati per sviluppare e testare sistemi di questo tipo [25] [26] [27] [28].

Ciò che accomuna tutti questi lavori è l'utilizzo di sensori di movimento come accelerometri e giroscopi: i primi misurano l'accelerazione lineare lungo i tre assi mentre i secondi l'accelerazione angolare. Valutare queste due grandezze fisiche è indispensabile quando si vuole analizzare e riconoscere il movimento di un corpo generico. Nel caso in questione, i dati forniti da questi sensori risultano particolarmente interessanti, perché si presume che una caduta determini una forte accelerazione in corrispondenza di uno o più dei tre assi di riferimento. Oggi, grazie al progresso della tecnologia MEMS (Micro Electro-Mechanical Systems), questo genere di sensori ha dimensioni molto ridotte e quindi sono facilmente installabili in uno spazio ristretto come nel caso di un dispositivo wearable.

Il problema principale nel rilevamento delle cadute è distinguerle da quelle attività quotidiane che potrebbero in qualche modo ingannare il software, come ad esempio quando un individuo si siede bruscamente su una sedia. Infatti, per sviluppare un algoritmo di questo tipo è necessario testarlo non solo con le cadute, ma anche con le attività quotidiane (camminare, salire le scale, sedersi, abbassarsi, sdraiarsi, ecc.) che potrebbero dare dei falsi positivi. A tal proposito, un aspetto molto importante che può influenzare molto l'efficienza dell'algoritmo è la posizione del dispositivo indosso al soggetto. Comunemente la scelta ricade sul polso, perché risulta la posizione più comoda, ma potrebbe anche essere indossato al braccio, alla vita oppure alla caviglia.

Alcuni approcci [28] ricorrono allo sviluppo di reti neurali in grado di riconoscere e distinguere diverse attività quotidiane come quelle sopra citate. Altri [26] [25], invece, si basano sulla semplice analisi del segnale proveniente dal sensore e controllano se i valori superano una certa soglia predefinita: queste applicazioni possiamo definirle "threshold-based". L'algoritmo proposto in questa tesi appartiene a quest'ultima categoria,

sebbene, come vedremo, non si limiti soltanto a controllare se viene superata o meno una certa soglia.

3.1.1 Utilizzo di un dataset per sviluppare un algoritmo

Lo sviluppo dell'algoritmo proposto in questa tesi è stato basato sui segnali registrati da un singolo accelerometro. In particolare, questi dati sono stati presi da un dataset che raccoglie numerose simulazioni sia di cadute che di attività quotidiane effettuate da più soggetti di differente età e sesso. Tale scelta, è motivata dall'indisponibilità di un ambiente adeguato e necessario per effettuare simulazioni rischiose come quelle delle cadute. Inoltre, si sarebbero dovute coinvolgere molte persone, possibilmente di costituzione corporea differente, al fine di avere molta varietà nei test. D'altra parte, per ottenere risultati più precisi è necessario testare l'algoritmo su quanti più test possibili, poiché, come vedremo a breve, i segnali ricavati da ogni caduta, seppur simili, non sono certamente uguali.

In rete sono disponibili alcuni dataset pubblici che possiedono le caratteristiche enunciate prima, ovvero che raccolgano sia test relativi a cadute che ad attività quotidiane e che coinvolgano soggetti di differente peso, altezza, età e sesso. Di seguito elenchiamo alcuni dataset che sono stati presi inizialmente in considerazione ma alla fine scartati per le ragioni che verranno esposte a breve; per semplicità da ora in poi identificheremo le attività quotidiane con la sigla ADL, ovvero Activity Daily Living.

- MobiFall [29]: 24 volontari tra i 22 e i 42 anni hanno partecipato al progetto, di cui 9 hanno simulato sia cadute che ADL e 15 soltanto cadute; il dataset contiene 9 tipi di ADL e 4 tipi di cadute registrate con uno smartphone Samsung Galaxy; ogni caduta è stata simulata tre volte.

- tFall [30]: 10 partecipanti tra i 20 e i 42 anni hanno simulato 8 tipi di cadute per un totale di 503 registrazioni con due smartphone; le ADL, invece, sono state registrate in modo continuo per una settimana intera.
- DLR [31]: 16 soggetti tra i 23 e i 50 anni hanno registrato 6 tipi di ADL; il numero e le condizioni delle cadute non sono definite dagli autori e i file risultano troppo brevi per effettuare analisi concrete.
- Project gravity [32]: 3 partecipanti di 22, 26 e 32 anni hanno eseguito 12 tipi di cadute e 7 tipi di ADL con uno smartphone posto in tasca.

Si nota subito che in questi dataset manca un dato importante, cioè il coinvolgimento di persone anziane, alle quali si vuole destinare l'applicazione in esame. Inoltre, gli ultimi due dataset dell'elenco non forniscono dati ed informazioni sufficienti per effettuare uno sviluppo corretto e preciso dell'applicazione. Infine, si ritiene poco preciso l'utilizzo di uno smartphone riposto in una tasca come dispositivo di registrazione. Si preferisce, invece, l'utilizzo di un dispositivo mirato indossato alla vita come fosse una cintura. Tale scelta è motivata dal fatto che, seguendo questa idea, si dovrebbero evitare alcuni falsi positivi che si otterrebbero, ad esempio, con un dispositivo indossato al polso, per via dei numerosi movimenti che il braccio esegue durante le comuni attività quotidiane.

Per tutte queste ragioni, la scelta è ricaduta sul SisFall dataset [33] di cui a breve vedremo tutti i dettagli e i vantaggi che esso offre rispetto ai precedenti. Tutte le informazioni riportate di seguito sono state estratte dall'articolo "*SisFall: A Fall and Movement Dataset*" [34].

Questo dataset contiene 15 tipi di simulazioni di cadute e 19 di ADL. Com'è possibile vedere nella tabella sottostante, sono state prese in considerazione molte circostanze in cui un soggetto potrebbe inavvertitamente cadere. Ogni simulazione ha una durata standard di 15 s ed è stata provata 5 volte per ottenere maggiore varietà.

Code	Activity	Trials	Duration
F01	Fall forward while walking caused by a slip	5	15 s
F02	Fall backward while walking caused by a slip	5	15 s
F03	Lateral fall while walking caused by a slip	5	15 s
F04	Fall forward while walking caused by a trip	5	15 s
F05	Fall forward while jogging caused by a trip	5	15 s
F06	Vertical fall while walking caused by fainting	5	15 s
F07	Fall while walking, with use of hands in a table to dampen fall, caused by fainting	5	15 s
F08	Fall forward when trying to get up	5	15 s
F09	Lateral fall when trying to get up	5	15 s
F10	Fall forward when trying to sit down	5	15 s
F11	Fall backward when trying to sit down	5	15 s
F12	Lateral fall when trying to sit down	5	15 s
F13	Fall forward while sitting, caused by fainting or falling asleep	5	15 s
F14	Fall backward while sitting, caused by fainting or falling asleep	5	15 s
F15	Lateral fall while sitting, caused by fainting or falling asleep	5	15 s

Tabella 3.1 - Tipi di cadute eseguite

Anche per le ADL sono state considerate molte attività, tra le quali alcune che potrebbero in qualche modo essere confuse come cadute, come la D11 che consiste nel provare ad alzarsi da una sedia per poi ricadere su di essa immediatamente.

Code	Activity	Trials	Duration
D01	Walking slowly	1	100 s
D02	Walking quickly	1	100 s
D03	Jogging slowly	1	100 s
D04	Jogging quickly	1	100 s
D05	Walking upstairs and downstairs slowly	5	25 s
D06	Walking upstairs and downstairs quickly	5	25 s
D07	Slowly sit in a half height chair, wait a moment, and up slowly	5	12 s
D08	Quickly sit in a half height chair, wait a moment, and up quickly	5	12 s
D09	Slowly sit in a low height chair, wait a moment, and up slowly	5	12 s
D10	Quickly sit in a low height chair, wait a moment, and up quickly	5	12 s
D11	Sitting a moment, trying to get up, and collapse into a chair	5	12 s
D12	Sitting a moment, lying slowly, wait a moment, and sit again	5	12 s
D13	Sitting a moment, lying quickly, wait a moment, and sit again	5	12 s
D14	Being on one's back change to lateral position, wait a moment, and change to one's back	5	12 s
D15	Standing, slowly bending at knees, and getting up	5	12 s
D16	Standing, slowly bending without bending knees, and getting up	5	12 s
D17	Standing, get into a car, remain seated and get out of the car	5	25 s
D18	Stumble while walking	5	12 s
D19	Gently jump without falling (trying to reach a high object)	5	12 s

Tabella 3.2 - Tipi di ADL eseguite

Per la raccolta di tutti questi dati sono stati coinvolti 38 partecipanti di cui 15 anziani (8 maschi e 7 femmine) e 23 giovani (11 maschi e 12 femmine). La tabella seguente mostra età, altezza e peso di ogni gruppo di soggetti.

	Sex	Age	Height (m)	Weight (kg)
Elderly	Female	62–75	1.50–1.69	50–72
	Male	60–71	1.63–1.71	56–102
Adult	Female	19–30	1.49–1.69	42–63
	Male	19–30	1.65–1.83	58–81

Tabella 3.3 - Età, altezza e peso dei partecipanti

Il gruppo dei soggetti giovani ha eseguito tutte le simulazioni sia per quanto riguarda le cadute che le ADL, mentre i soggetti anziani, per motivi di salute, hanno eseguito soltanto le ADL ad eccezione delle D06, D13, D18 e D19. Inoltre, a causa di particolari motivi di salute personali, alcuni soggetti anziani non hanno eseguito tutte le ADL previste.

Infine, per registrare i dati è stato utilizzato un dispositivo assemblato dal team di sviluppo ed equipaggiato con tre sensori: un accelerometro AXL345, un accelerometro MMA8451Q e un giroscopio ITG3200. Di questi, ai fini dello sviluppo dell'algoritmo è stato utilizzato soltanto il primo. In figura è possibile vedere la disposizione e l'orientamento del dispositivo indosso ad un generico soggetto. Tutti i dati sono stati campionati a 200Hz. Per ulteriori dettagli si rimanda all'articolo [34].

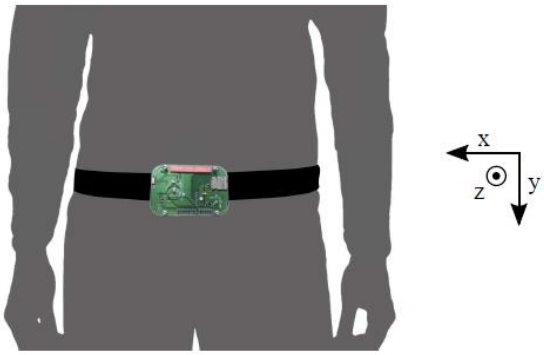

Figura 3.1 - Orientamento e disposizione del dispositivo indosso ad un soggetto

3.1.2 Sviluppo dell'algoritmo

L'algoritmo proposto in questa tesi appartiene alla categoria threshold-based sopra citata, sebbene, come vedremo, non si limiti soltanto a controllare se viene superata o meno una certa soglia.

Lo sviluppo è iniziato analizzando il segnale, o per meglio dire, i segnali registrati dall'accelerometro durante la simulazione di una generica caduta. Ogni test è caratterizzato da tre segnali corrispondenti ai tre assi di riferimento. Si è notato come tutti questi segnali siano caratterizzati da un picco più o meno intenso in corrispondenza della caduta, come d'altronde era facile aspettarsi.

Dunque, il primo approccio, nonché quello più semplice, è stato quello di selezionare un valore soglia e verificare quanti test venivano riconosciuti come cadute. Questo valore deve essere sufficientemente alto in modo da non rilevare anche semplici movimenti bruschi come delle cadute. D'altra parte, se si imposta un valore troppo alto si rischia di non rilevare gran parte delle cadute effettive, perché purtroppo i valori dei picchi sono spesso molto diversi tra loro e ciò significa che non esiste una soglia comune che li abbracci tutti. Di seguito si riportano i grafici di alcune differenti cadute: l'asse x riporta il numero di campioni, mentre l'asse y i valori grezzi forniti dall'accelerometro (più avanti vedremo come è possibile convertire questi dati in una grandezza più appetibile).

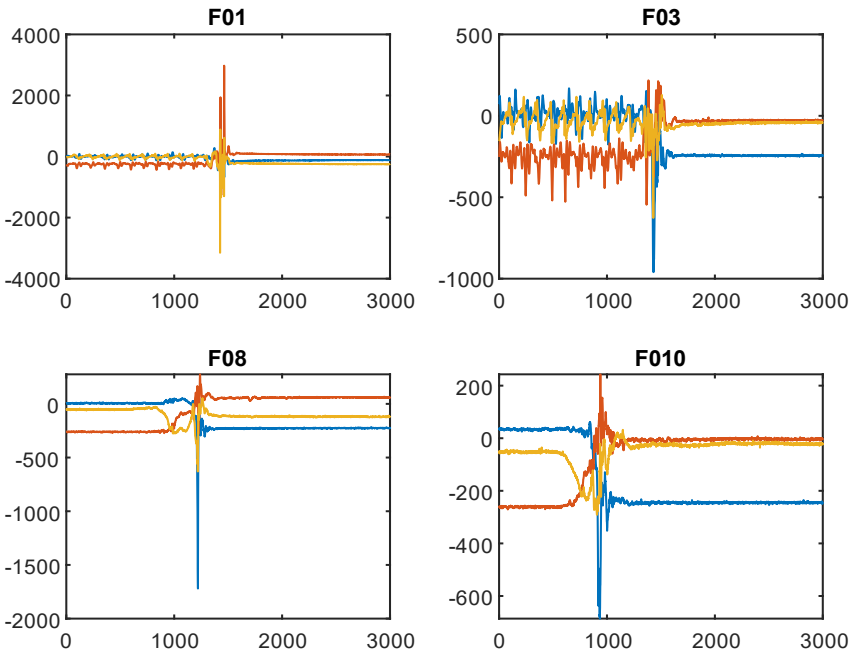

Figura 3.2 - Esempio grafici di quattro cadute differenti

Com'è possibile notare i valori di picco variano molto tra una caduta ed un'altra. Inoltre, si nota subito un particolare: il segnale relativo ad ogni asse segue un andamento che dipende dall'orientamento dell'asse stesso, ciò significa che i valori sono influenzati dalla componente gravitazionale. Dunque, quando il sensore è fermo, il valore relativo ad ogni asse non si assesta ad un valore prossimo a zero, ma al valore di accelerazione gravitazionale che subisce in base all'orientamento dell'asse stesso. Sarebbe quindi interessante non solo eliminare la componente di gravità, ma anche far riferimento ad un unico segnale che sia la sintesi dei tre.

A questo punto si è scelto di far riferimento alla norma dei tre segnali al quale viene sottratta la componente di gravità. Ciò significa che per ogni terna di valori (x, y e z) viene calcolata la norma come segue:

$$norm = \sqrt{x^2 + y^2 + z^2}$$

Di seguito si riporta il grafico della norma relativa al test F01: com'è possibile notare, la forma della curva non è stata alterata e il picco è rimasto molto evidente.

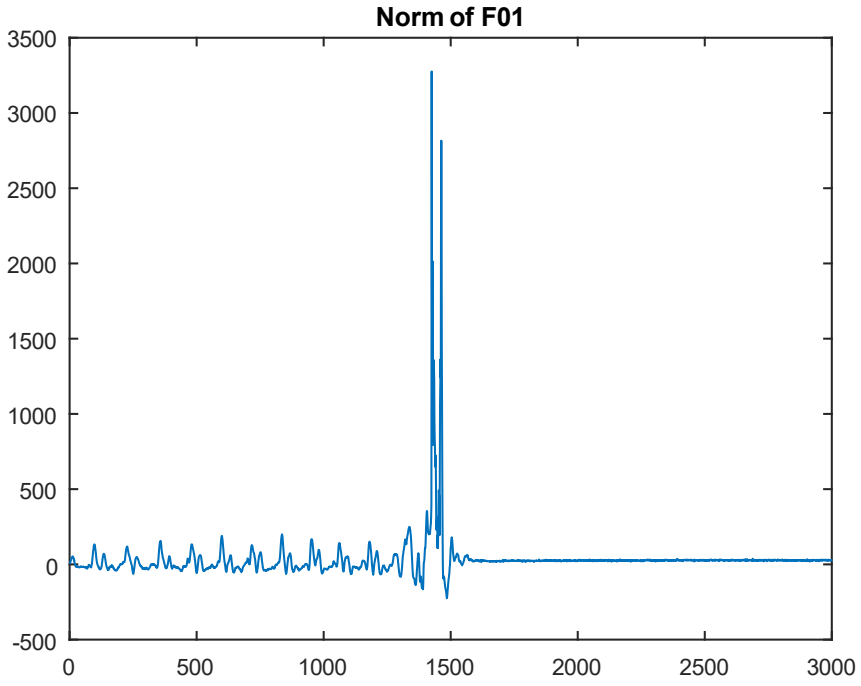

Figura 3.3 - Norma dei segnali x, y e z relativi a F01

Gli autori di [34] hanno scelto di filtrare in frequenza i segnali al fine di ripulirli ed ottenere risultati più accurati. Questa operazione è stata testata anche in questa sede utilizzando sia filtri passa-alto che filtri passa-banda di Butterworth. Ciò che si è ottenuto è semplicemente un'attenuazione del segnale e provando un semplice algoritmo threshold-based, i risultati non erano diversi da quelli ottenuti a partire dai segnali grezzi.

A questo punto, si è pensato di complicare un po' l'algoritmo, seguendo una logica più mirata, dal momento che una semplice funzione che controlla se viene superato un valore soglia è chiaramente troppo riduttiva. Valutando i segnali relativi a molteplici test, si è visto che dopo il valore di picco, la curva si assesta ad un valore più o meno costante, dovuto al fatto che il soggetto si trova sdraiato a terra e quindi fermo. Questa informazione è stata ritenuta molto utile ai fini del rilevamento di una caduta.

Seguendo questa considerazione, si è pensato di valutare la varianza in due finestre temporali consecutive: la prima centrata nel valore che ha superato la soglia e la seconda immediatamente successiva alla prima. Ciò che ci si aspetta è che la varianza centrata nel picco sia molto alta a causa dell'urto provocato dalla caduta, mentre la varianza relativa ai campioni successivi sia molto bassa poiché, come detto prima, si presume che il soggetto sia a terra e quindi i valori dell'accelerometro siano approssimativamente costanti. In questo modo, si vogliono scartare tutte quelle ADL che, a causa di qualche movimento leggermente brusco, possono essere scambiate erroneamente per una caduta. Per chiarire meglio questo concetto, di seguito viene illustrato un esempio che mette a confronto due segnali relativi ad una caduta e ad una ADL che potrebbe ingannare l'algoritmo.

Com'è possibile notare nel grafico della ADL D05, molti campioni raggiungono valori piuttosto alti al punto da superare un'eventuale soglia predefinita ma comunque preso uno qualsiasi di questi campioni, i

successivi continuano a variare molto determinando una varianza abbastanza alta; per tale ragione questa attività non sarà identificata come caduta, come è giusto che sia. Differente andamento ha invece la curva della caduta F03, la quale è caratterizzata da un forte picco seguito da un andamento pressoché costante.

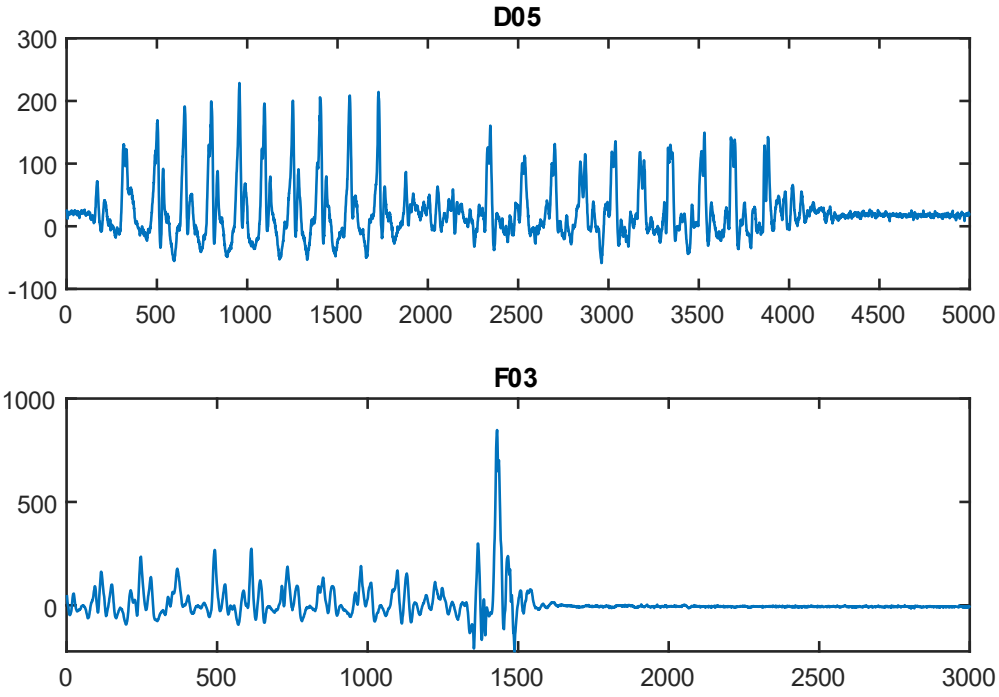

Figura 3.4 - Due segnali messi a confronto. D05 è il segnale registrato mentre si salgono le scale, mentre F03 è relativo ad una caduta laterale.

Dopo aver illustrato l'idea alla base dell'algoritmo vediamo adesso lo sviluppo pratico e infine i risultati ottenuti. Tutti i test e gli esperimenti che vedremo sono stati effettuati su MATLAB per poter lavorare facilmente con la moltitudine di test contenuti nel dataset.

È bene precisare che, per valutarne l'efficienza, l'algoritmo è stato applicato da un lato all'insieme dei test delle cadute e dall'altro all'insieme delle ADL; in questo modo è stato possibile controllare come le modifiche apportate influenzavano entrambi i fronti. Precisiamo, inoltre che l'insieme delle cadute è stato utilizzato per intero, mentre l'insieme delle ADL è stato

preso facendo riferimento soltanto ai test effettuati dalle persone anziane, poiché sono le dirette interessate di un'applicazione di questo tipo. Ricordiamo, come detto in precedenza, che alcune delle attività elencate in Tabella 3.2 non sono state svolte dai soggetti anziani per motivi di salute. Per semplicità da qui in avanti identificheremo l'insieme delle cadute con "A" e l'insieme delle ADL con "B".

Illustriamo di seguito in maniera schematica gli step principali che compongono l'algoritmo:

1. Per ogni campione controlla se il valore di quest'ultimo supera una soglia predefinita (THR – threshold);
2. In caso positivo calcola due valori:
    a. La varianza della finestra di campioni centrata su quello che ha superato la soglia (VarP - variance peak);
    b. La varianza della finestra successiva alla precedente (VarA – variance after);
3. Se VarP è maggiore di un'altra soglia predefinita THRP e VarA è minore di un'altra soglia THRA, allora l'attività viene identificata come caduta.

Si nota subito come in questa versione finale dell'algoritmo siano emerse due nuovi valori di soglia, necessari per valutare i casi limite. L'efficienza dell'algoritmo è quindi determinata da tre valori di soglia, che devono essere opportunamente scelti affinché le cadute vengano identificate come tali e viceversa per le ADL. Ci aspettiamo che i valori di VarA relativi alle cadute si mantengano piuttosto bassi rispetto a quelli relativi alle ADL. Discorso inverso vale per i valori di VarP, i quali dovrebbero mantenersi più alti per le cadute, dal momento che esse dovrebbero comportare intense variazioni in prossimità del picco.

Si riporta di seguito uno pseudo codice che sintetizza il funzionamento dell'algoritmo per ogni test.

```
%% constants
w_size = 500;
thr = 400;
thrP = 7000;
thrA = 550;

for sample in test
   if sample > thr
          index = get_index(sample,test);
      varP = var(sample(index-w_size/2:index+w_size/2));
      varA = var(sample(index+w_size/2:index+(3*w_size/2)));
      if (varP>thrP && varA<thrA)
         disp('FALL DETECTED\n');
      end
   end
end
```

Iniziamo col descrivere il primo blocco di codice dove sono definite alcune costanti. La prima definisce la larghezza delle finestre all'interno del quale saranno calcolate le varianze di cui si è parlato più volte in precedenza. Questo valore è stato scelto sperimentalmente e dipende moltissimo dalla frequenza di campionamento al quale i dati vengono registrati: più è alta tale frequenza, più campioni si ottengono e quindi più grande deve essere la finestra. Ricordiamo che in questo caso la frequenza di campionamento è di 200 Hz e la durata di ciascun test è di 15 s, quindi risultano 3000 campioni.

In realtà, come si vede dal codice, la larghezza della finestra successiva al picco è stata impostata al doppio di w_size. Tale scelta si basa sul fatto che immediatamente dopo l'impatto a terra, il soggetto effettua dei movimenti di minore intensità e di breve durata prima di essere del tutto fermo a terra. A seconda della durata di questi movimenti, si rischia che le relative oscillazioni del segnale non ricadano nella finestra centrata nel picco; di conseguenza se la finestra successiva fosse troppo piccola, la varianza ad essa associata verrebbe influenzata troppo da queste oscillazioni e risulterebbe troppo alta. In altri termini, vogliamo che tale

finestra racchiuda quella porzione di segnale approssimativamente costante che abbiamo visto prima nei grafici di esempio Figura 3.2.

In generale si consiglia di scegliere una finestra anche più del doppio di w_size; in questo contesto, purtroppo si era vincolati dal tempo e quindi dal numero fisso di campioni, per cui se la caduta fosse avvenuta in prossimità della fine del test, si sarebbe rischiato di sforare, dal momento che lo stesso algoritmo con gli stessi parametri veniva applicato a tutto l'insieme A.

Il secondo parametro, già menzionato pocanzi, è THR, ovvero quel valore di soglia che fa scattare l'algoritmo effettivo. Il suo valore, pari a 400, è stato calcolato facendo riferimento all'insieme A, cercando di rilevare come cadute quanti più casi (test) possibili, ma allo stesso tempo senza coinvolgere test dell'insieme B.

Per determinare, invece, i valori di THRP e THRA si è proceduto come segue. È stato fatto eseguire l'algoritmo saltando il punto 3 (di cui sopra) e sono state memorizzati i valori di VarP e VarA in due array. Questa operazione è stata svolta su entrambi gli insiemi A e B distintamente. Per quanto riguarda i valori di VarA relativi all'insieme A, si è visto che il valore massimo era 540, ad eccezione di due valori che superavano l'ordine di 10^3. Mentre, per l'insieme B, il valore minimo era di 570, ad eccezione però di 10 valori che erano inferiori a 30. Dopo queste osservazioni, il valore di THRA è stato impostato a 550, ossia un valore intermedio tra i due citati pocanzi. Le considerazioni fatte per VarP sono simili a quelle fatte per VarA, sebbene i valori ottenuti per entrambi gli insiemi siano pressoché dello stesso ordine di grandezza. Ciò che infatti è determinante per discriminare le attività dei due insiemi è infatti la condizione relativa a VarA. Da un punto di vista sperimentale si è visto che un valore di VarP pari a 7000 risulta ottimale per in nostri scopi.

Ovviamente i valori di soglia ottenuti sono validi soltanto se l'algoritmo viene applicato ai test di questo specifico dataset. Il motivo sta nel fatto che

ogni accelerometro fornisce i dati secondo una propria scala di misura e i valori di varianza e di soglia sono stati calcolati a partire proprio da questi dati. In realtà nel datasheet di ogni accelerometro è riportato un fattore di conversione che, moltiplicato per il dato grezzo, fornisce la misura in termini di accelerazione gravitazionale. Questo fattore si calcola come segue.

$$F = \frac{2 * Range}{2 * Resolution}$$

Il range corrisponde al fondo scala impostato in fase di inizializzazione del sensore e si misura in g; ad esempio, nel caso in questione questo valore è ±16g, ma potrebbe anche essere impostato a ±2g o ±8g. La risoluzione, invece, identifica quanti bit sono riservati per contenere il valore registrato; nel caso del AXL345 la risoluzione è 13 bit.

Per valutare le performance dell'algoritmo sono stati calcolati tre indici: sensibilità (SE), specificità (SP) e accuratezza (AC). Di seguito si riportano le formule:

$$SE = \frac{TP}{TP + FN} \qquad SP = \frac{TN}{TN + FP} \qquad AC = \frac{SE + SP}{2}$$

TP e TN sono rispettivamente il numero di veri positivi (true positives) e di veri negativi (true negatives), mentre FP ed FN sono il numero di falsi positivi (false positives) e di falsi negativi (false negatives). AC (accuracy) è semplicemente la media tra SE (sensitivity) e SP (specificity).

Per calcolare l'indice SE, dal momento che le informazioni desiderate sono i veri positivi e i falsi negativi, è stato fatto eseguire l'algoritmo sull'insieme A conteggiando quanti test sono stati rilevati come cadute. Analogamente si è proceduto per calcolare l'indice SP, con la differenza che l'insieme in esame in questo caso era B. I risultati ottenuti con i valori di soglia precedentemente esposti sono i seguenti:

$$SE = 96.58 \qquad SP = 98.18 \qquad AC = 97.38$$

I risultati dimostrano una buona efficienza dell'algoritmo trattato. Gli autori di [34], valutando anch'essi la norma del segnale ma considerando soltanto l'asse x e z, hanno ottenuto un valore di AC pari a circa 90.45. È opportuno riportare che il loro approccio è basato sulla ricerca di un unico valore di soglia che viene applicato a diverse caratteristiche del segnale, come ad esempio l'orientamento degli assi, il modulo della deviazione standard e tra questi anche la norma. Il massimo valore di AC che hanno ottenuto, corrispondente al modulo della deviazione standard, è di 92.58. Gli autori non hanno di fatto proposto un algoritmo, bensì un'analisi e una valutazione dei segnali ricavati dai loro esperimenti facendo riferimento ad un valore di soglia ricavato in corso d'opera. Per tutti i dettagli, che ovviamente non riporteremo in questo scritto, si rimanda all'articolo sopra citato.

Osservando i risultati ottenuti, si nota come l'indice SP sia leggermente superiore all'indice SE: tale differenza non è casuale. Durante la ricerca dei valori di soglia ottimali, si è visto come, variando questi ultimi, o aumentava SE e diminuiva SP o viceversa. Dunque, l'obiettivo era trovare un giusto compromesso tra i due indici, ovvero far sì che venissero rilevate quante più cadute possibili come tali e allo stesso tempo venissero escluse quante più ADL possibili. Infine, come abbiamo visto, sono stati scelti i valori di soglia discussi pocanzi e si è preferito dare maggior peso all'indice SP, seppur con una lieve differenza di 0.2 dall'indice SE. Tale scelta è motivata dal fatto che le cadute prese in esame sono ovviamente delle simulazioni ed è plausibile pensare che alcuni soggetti in alcuni test abbiano in qualche modo attenuato l'urto a terra, come d'altronde verrebbe istintivo ad ognuno di noi. Una simulazione, per quanto possa essere fedele non sarà mai identica al fenomeno reale, ancor più in un contesto di questo tipo. Infatti, ciò che principalmente contraddistingue una caduta simulata da una reale è la consapevolezza che il soggetto ha in fase di simulazione.

Nel prossimo paragrafo vedremo la piattaforma hardware che è stata scelta per simulare il dispositivo wearable al fine di testare il sistema insieme al dispositivo fisso, come vedremo successivamente nel quarto capitolo.

3.2 Piattaforma hardware

Negli ultimi anni, il mercato delle schede di sviluppo è cresciuto moltissimo, basti pensare al successo riscontrato da schede come Arduino e RaspberryPi. Queste piattaforme hardware dalle dimensioni piuttosto contenute dispongono solitamente di un'interfaccia fisica alla quale è possibile collegare sensori, attuatori, display e altri componenti hardware esterni. Il fine è quello di creare il prototipo di un dispositivo in grado di leggere ed elaborare informazioni interne e/o esterne, eseguire determinate azioni attraverso alcuni attuatori (se presenti) o inviare i dati ad un altro dispositivo attraverso un modulo di comunicazione wired o wireless.

Per il progetto di questa tesi è stata scelta la SensorTile (STEVAL-STLCS01V1) [35], una scheda di sviluppo prodotta dalla STMicroelectronics. A differenza delle sue simili, questa scheda si contraddistingue per le sue dimensioni molto ridotte e per avere già a bordo diversi sensori, risultando la scelta ideale in ambito wearable. Infatti, in una superficie di soli 13.5 x 13.5 mm, sono integrati sensori ambientali (temperatura, pressione) sensori di movimento (accelerometro, giroscopio, magnetometro), un microfono e un modulo Bluetooth Low Energy (BLE)

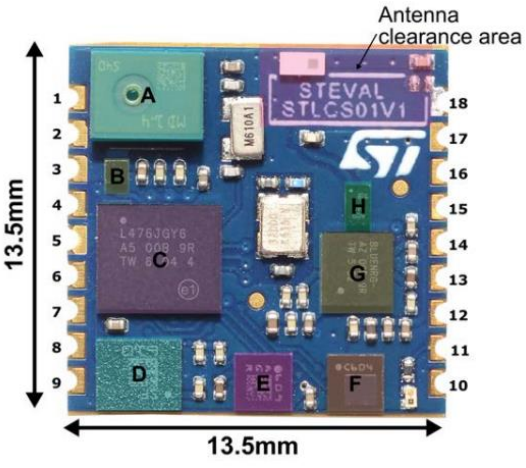

Figura 3.5 - STEVAL-STLCS01V1 [43]

insieme con un microcontrollore STM32. Di seguito si riporta una tabella riassuntiva elencante tutti i componenti a bordo.

Reference	Device	Description
A	MP34DT05-A	MEMS audio sensor digital microphone
B	LD39115J18R	150 mA low quiescent current low noise LDO 1.8 V
C	STM32L476 MCU	ARM Cortex-M4 32-bit microcontroller
D	LSM6DSM	iNEMO inertial module: low-power 3D accelerometer and 3D gyroscope
E	LSM303AGR	Ultra-compact high-performance eCompass module: ultra-low power 3D accelerometer and 3D magnetometer
F	LPS22HB	MEMS nano pressure sensor: 260-1260 hPa absolute digital output barometer
G	BlueNRG-MS	Bluetooth low energy network processor
H	BALF-NRG-02D3	50 Ω balun with integrated harmonic filter

Figura 3.6 - Componenti principali della SensorTile [43]

Il cuore della SensorTile è rappresentato dal STM32L476JGY, un microcontrollore ARM Cortex-M4 a 32 bit da 80MHz a bassissimo consumo. Esso integra una memoria RAM da 128KB ed una memoria Flash da 1MB. Considerando le ridotte dimensioni della scheda si può dire che essa offre risorse hardware piuttosto notevoli, al punto che i pacchetti software forniti dall'azienda implementano applicazione non proprio banali, come quelle per il riconoscimento delle attività, riconoscimento della postura e riconoscimento di gesture.

Come accennato prima questa scheda dispone di diversi sensori a bordo. Si riportano di seguito, in maniera schematica, tutti i dettagli tecnici di ciascuno di essi, tratti sempre dalla documentazione ufficiale sopra citata.

- LSM6DSM [36]:
    - un modulo che integra un accelerometro 3D e un giroscopio 3D;
    - consumo pari a 0.4 mA in "normal mode" e 0.65 mA in "high-performance mode";

    - funzione “always-on” che garantisce il monitoraggio continuo senza influire eccessivamente sul consumo energetico;
    - fondo scala di ±2/±4/±8/±16 g per l’accelerometro e di ±125/±250/±500/±1000/±2000 dps per il giroscopio;
    - coda FIFO di 4KB per memorizzare i valori;
    - i campioni vengono memorizzati a 16 bit in due registri da 1 byte;
    - interfaccia seriale SPI e I2C.
- LSM303AGR [37]:
    - un modulo che integra un accelerometro 3D ed un magnetometro;
    - fondo scalda di ±2/±4/±8/±16 g per l’accelerometro e di of ±50 gauss per il magnetometro;
    - interfaccia SPI e I2C con supporto a quattro modalità: standard (100KHz), fast mode (400KHz), fast mode plus (1MHz) e high-speed (3.4MHz);
    - i valori sono memorizzati a 16 bit in una coda FIFO.
- LPS22HB [38]:
    - un modulo che integra un sensore piezoresistivo in grado di misurare la pressione ed un sensore di temperatura;
    - funziona come barometro.
    - interfacce I2C ed SPI;
    - il range di valori di pressione misurabili è tra i 260 e i 1260 hPa;
    - frequenza di campionamento regolabile a partire da 1 Hz fino a 75 Hz;
    - i valori di pressione sono memorizzati a 26 bit, mentre quelli di temperatura a 16 bit;

- anche in questo caso i dati sono memorizzati in una coda FIFO;
- consumo pari 3 μA, risultando davvero ideale per dispositivi alimentati a batteria.

Il BlueNRG-MS [39] è un modulo Bluetooth Low Energy basato sulle specifiche Bluetooth 4.2. Esso integra un microcontrollore ARM Cortex-M0 a 32 bit nel quale è implementato l'intero stack Bluetooth. Quest'ultimo è del tutto riprogrammabile al fine di creare i propri servizi e caratteristiche GATT/ATT di cui parleremo a breve. Il modulo è collegato al microcontrollore della SensorTile attraverso un bus SPI a 4 canali attraverso il quale transitano i messaggi in entrata e in uscita. Inoltre, questo modulo integra anche un coprocessore che permette la cifratura AES dei messaggi.

Qui di seguito si riporta uno schema che illustra l'interconnessine tra tutti i componenti della SensorTile. È possibile notare anche i bus utilizzati per connettere ciascuna periferica.

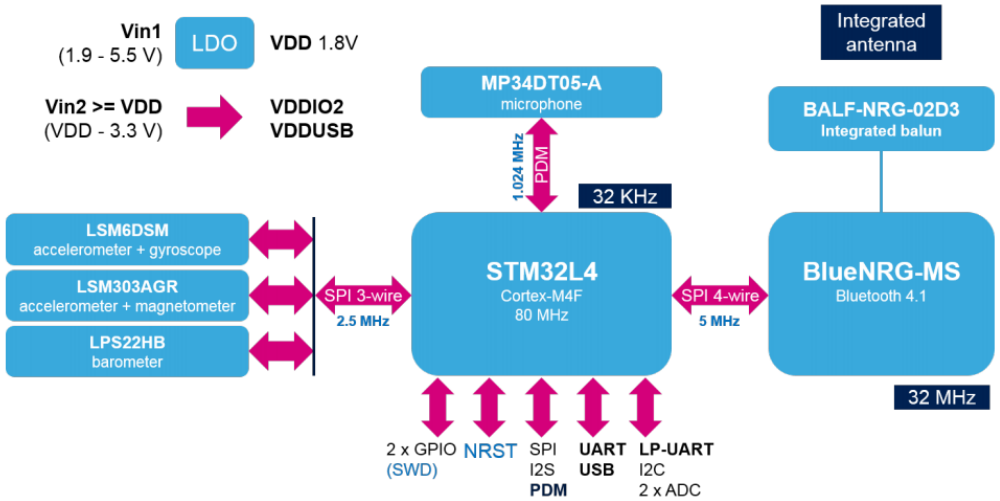

Figura 3.7 - Schema a blocchi funzionale della SensorTile [43]

Dal momento che, come si sarà facilmente intuito, la comunicazione tra il nodo fisso e il nodo wearable del sistema sarà basata sul BLE, si ritiene opportuno analizzare in dettaglio tutti gli aspetti di questo protocollo.

3.3 Bluetooth Low Energy

Come accennato prima, il BLE opera alla stessa banda di frequenze radio del Bluetooth classico [19], ovvero tra 2.4 e 2.4835 GHz, ma adopera una diversa gestione dei canali. Il Bluetooth classico utilizza 79 canali da 1MHz, mentre il BLE ne utilizza 40 da 2MHz. La trasmissione dei dati lungo i canali è garantita ad un bit rate di 1 Mbit/s (con un'opzione di 2 Mbit/s nel Bluetooth 5) e la potenza di trasmissione massima è di soli 10 mW (100 mW nel Bluetooth 5), risultando quindi molto "low energy" rispetto alla versione classica.

Per rendere meglio l'idea dell'efficienza energetica di BLE citiamo uno studio condotto da una azienda software dal nome Aislelabs [40], la quale ha riferito che le periferiche come i Beacon di prossimità, di solito funzionano per 1-2 anni con una batteria a moneta da 1.000 mAh. Ovviamente questo risultato non vale per tutti i dispositivi del mondo, poiché l'efficienza è dettata anche dal chipset del dispositivo, dal tipo di scansione (continua o discreta) e infine anche dal sistema operativo a bordo. Per esempio, negli ultimi anni i produttori di dispositivi Android e iOS hanno ottimizzato parecchio il consumo energetico, rendendo l'utilizzo quotidiano di BLE quasi trascurabile.

Vediamo adesso come funziona il procedimento di annuncio e scoperta dei dispositivi BLE, ovvero come essi comunicano la loro disponibilità di connessione e come gli altri dispositivi riescono a identificarli e ad instaurare una comunicazione con essi. Quando un dispositivo BLE viene messo in ascolto, ovvero in attesa che un altro dispositivo instauri una comunicazione con esso, inizia a trasmettere una serie di pacchetti di annuncio. Più precisamente questi pacchetti transitano su tre canali distinti, al fine di ridurre le interferenze. I pacchetti vengono inviati in sequenza su almeno uno di questi tre canali secondo una certa cadenza, chiamata

intervallo di annuncio, per evitare collisioni sullo stesso canale. L'intervallo di annuncio viene dettato da un ritardo casuale (fino a 10 millisecondi) tra una trasmissione e un'altra sullo stesso canale.

Dall'altro lato, invece, un dispositivo ascolta il canale per una durata chiamata finestra di scansione, che viene periodicamente ripetuta ad ogni intervallo di scansione. La latenza di scoperta è quindi determinata da un processo probabilistico e dipende dai tre parametri, cioè l'intervallo di annuncio, l'intervallo di scansione e la finestra di scansione.

3.3.1 Modello Software

Le specifiche BLE definiscono diversi profili standard che indicano il modo in cui un dispositivo funziona in una data applicazione, ovvero quali dati devono transitare e quali regole devono seguire [19]. Ogni dispositivo può implementare uno o più profili. Per esempio, un profilo potrebbe essere dedicato al monitoraggio del battito cardiaco, un altro per la temperatura corporea e un altro ancora per il livello della batteria. I produttori di dispositivi BLE possono decidere di seguire i profili standard oppure crearne altri personalizzati; il protocollo sottostante garantirà comunque la compatibilità fra i dispositivi, che saranno ugualmente in grado di leggere i profili personalizzati. Di seguito si riporta uno schema dello stack BLE dove è possibile vedere la collocazione del protocollo GATT/ATT.

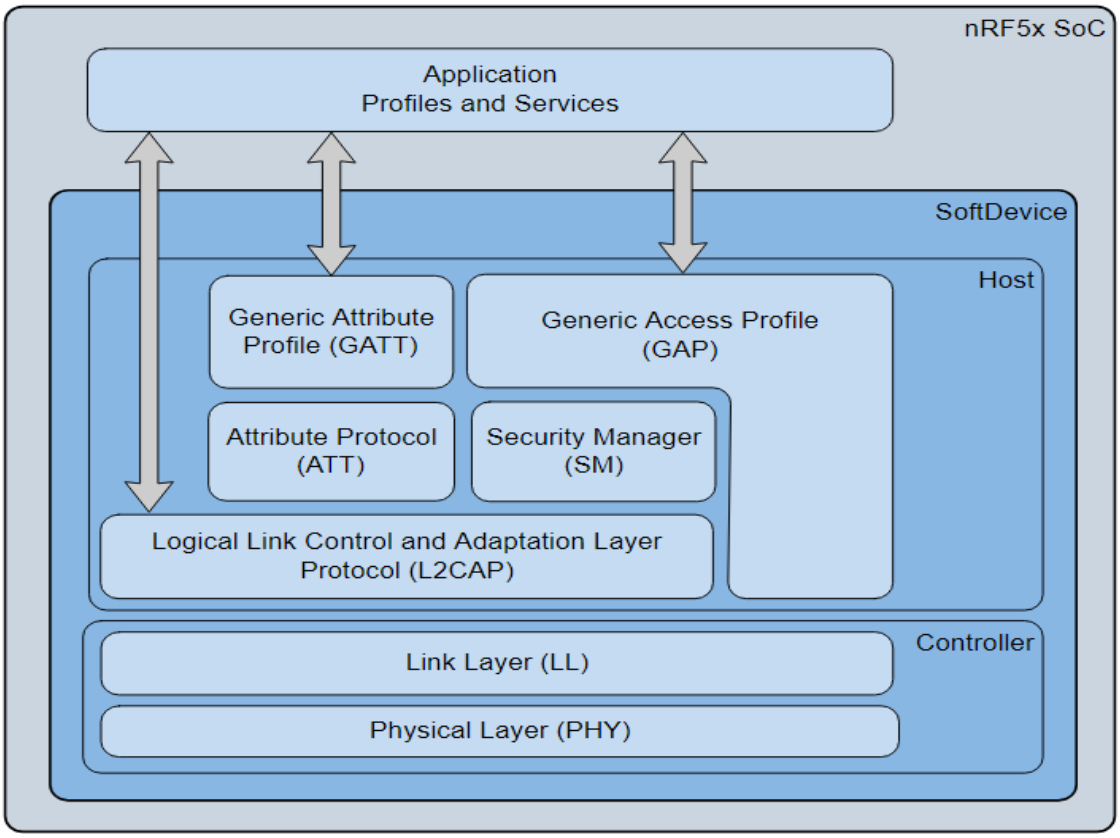

Figura 3.8 - Stack protocollare del BLE [45]

La maggior parte dei profili implementati nei dispositivi che utilizzano BLE si basa sul Generic Attribute Profile (GATT), il quale definisce una API (Application Programming Interface) per l'invio e la ricezione di pacchetti di dati noti come attributi. GATT a sua volta di basa su ATT (Attribute Protocol), infatti spesso ci si riferisce al protocollo GATT/ATT. Questo protocollo è stato sviluppato appositamente per il BLE e quindi non è supportato dal Bluetooth classico.

Vediamo adesso più in dettaglio la terminologia di GATT/ATT.

- **Client**: un dispositivo che avvia comandi e richieste GATT e accetta risposte, ad esempio, un computer o uno smartphone.
- **Server**: un dispositivo che riceve comandi e richieste GATT e restituisce risposte, ad esempio, un dispositivo dotato di un sensore di temperatura.
- **Caratteristica**: un valore di dati trasferito tra client e server, ad esempio, l'attuale valore di temperatura.

- **Servizio**: una raccolta di caratteristiche correlate da un significato comune, che operano insieme per eseguire una funzione particolare. Ad esempio, il servizio *Health Thermometer* include le caratteristiche per un valore di misurazione della temperatura e un intervallo di tempo tra le misurazioni.
- **Descrittore**: un descrittore fornisce informazioni aggiuntive su una determinata caratteristica. Ad esempio, la caratteristica del valore di temperatura può avere un'indicazione delle sue unità di misura (ad esempio Celsius o Fahrenheit) e i valori massimo e minimo che il sensore può misurare. I descrittori sono opzionali e ogni caratteristica può avere uno o più descrittori.

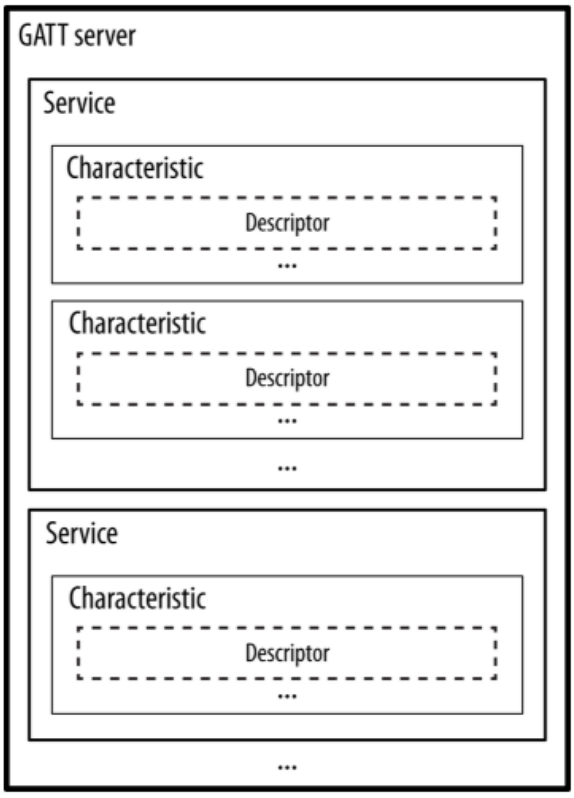

Figura 3.9 - Gerarchia dei dati GATT [44]

Qui a fianco si riporta un semplice schema che evidenzia le relazioni tra profilo (GATT server), servizio e caratteristiche.

Alcuni valori dei servizi e delle caratteristiche vengono utilizzati per scopi amministrativi: ad esempio, il nome del dispositivo e il numero di serie possono essere letti come caratteristiche standard all'interno del servizio Generic Access. Analogamente con quanto avviene con le funzioni, i servizi possono includere altri servizi: si vengono così a creare i servizi primari e i servizi secondari.

Servizi, caratteristiche e descrittori sono indicati collettivamente come attributi e ciascuno di essi è identificato da un identificatore noto come UUID, un valore a 128 bit. Gli UUID possono essere standard o custom.

Nel caso di quelli standard, la specifica prevede uno specifico intervallo nel formato *xxxxxxxx-0000-1000-8000-00805F9B34FB* e per questioni di efficienza e di praticità, questi identificatori sono rappresentati come valori a 16 o 32 bit, piuttosto che a 128 bit come richiesto per un UUID completo. Ad esempio, il servizio *Device Information*, previsto dallo standard e implementato in qualsiasi applicazione BLE, ha il codice breve 0x180A, anziché 0000180A-0000-1000-…. Una lista completa di tutti gli attributi standard con relativo UUID è disponibile online sul sito ufficiale del protocollo [41]. Nel caso degli UUID custom, i programmatori di applicazioni BLE possono scegliere un valore a 128 bit casuale.

Infine, abbiamo un ulteriore valore noto come handle che funge da identificatore. L'handle è sostanzialmente un numero progressivo a 16 bit che serve a identificare univocamente servizi, caratteristiche e descrittori all'interno di un profilo.

Di seguito viene riportato, come esempio, il servizio GATT standard dedicato allo scambio di dati relativi al battito cardiaco.

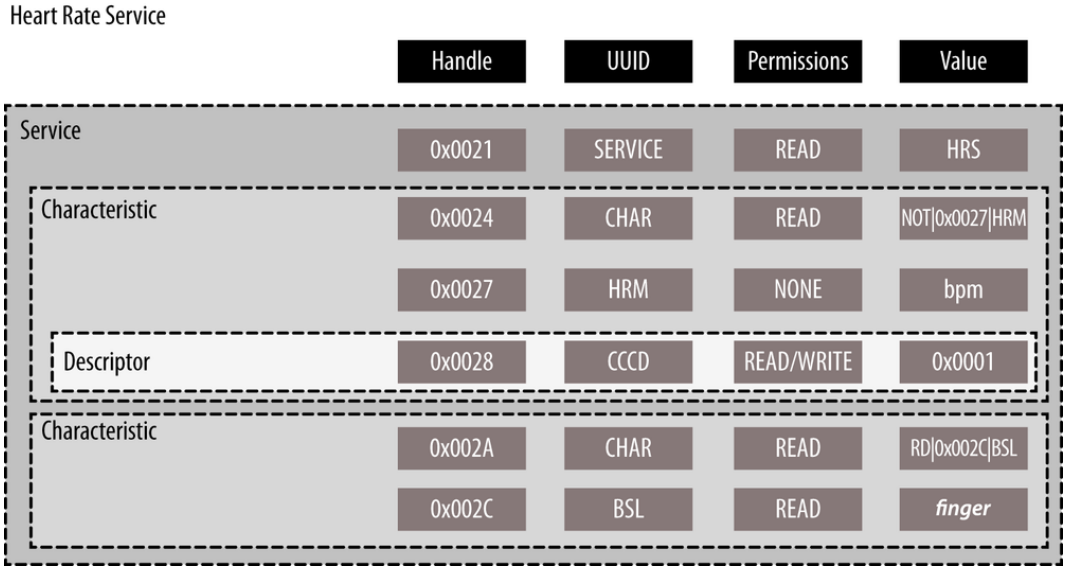

Figura 3.10 - Servizio GATT standard per Heart Rate [44]

3.3.2 Operazioni GATT

Il protocollo GATT, come accennato prima, fornisce una API di alto livello che permette al client di svolgere diverse operazioni con il server [19]. Ad esempio, attraverso un set di metodi il client può recuperare tutte le informazioni del server, ovvero può eseguire le seguenti operazioni:

- Scoprire l'UUID per tutti i servizi primari;
- Trovare un servizio con un dato UUID;
- Trovare servizi secondari per un determinato servizio primario;
- Scoprire tutte le caratteristiche di un determinato servizio;
- Trovare una caratteristica a partire da un dato UUID;
- Leggere tutti i descrittori di una particolare caratteristica;

Esistono poi altri metodi che permettono di eseguire operazioni di READ, WRITE, NOTIFY e INDICATION, spesso note come permessi, in quanto stabiliscono a tutti gli effetti i permessi concessi al client circa il valore di una caratteristica.

Le operazioni di READ possono essere eseguite o specificando l'UUID della caratteristica oppure specificando un handle. Quest'ultimo può essere letto attraverso i metodi sopra citati. Le operazioni di WRITE si effettuano sempre a partire dall'handle ed è possibile specificare se è richiesta o meno la risposta dal server; a tal proposito esiste la variante WRITE WITHOUT RESPONSE. Esistono anche due varianti, ovvero LONG READ e LONG WRITE, utili quando la lunghezza dei dati supera quella del MTU (Maximun Transmission Unit), solitamente di 20 bytes.

L'operazione di NOTIFY indica invece quando un client può richiedere, per l'appunto, una notifica al server ogni qual volta il valore di una determinata caratteristica cambia. Ad esempio, un server sensore di temperatura può notificare al proprio client ogni volta che prende una misurazione. Ciò evita la necessità per il client di eseguire il polling sul

server, il che richiederebbe che il circuito radio del server fosse costantemente operativo, con conseguente dispendio energetico.

L'operazione di INDICATION è molto simile a quella di NOTIFY, tranne per il fatto che richiede una risposta dal client, come conferma di ricezione del messaggio.

Nel prossimo e ultimo capitolo sarà affrontata la virtualizzazione dei sensori, già citata in precedenza, nel quale saranno coinvolti un dispositivo wearable e uno fisso al fine di testare una generica applicazione distribuita.

4 Virtualizzazione di sensori

Nel capitolo precedente abbiamo visto lo sviluppo di un'applicazione orientata all'health monitoring ed in particolare al rilevamento delle cadute, analizzando tutte le fasi di sviluppo di uno specifico algoritmo e infine valutandone l'accuratezza. In questo capitolo, invece, si approfondirà la fase di test del sistema insieme al dispositivo fisso. Come già accennato all'inizio del capitolo precedente, il tema principale che è stato affrontato è la virtualizzazione dei sensori: è giunto il momento di spiegare di cosa si tratta.

La ricerca moderna, come abbiamo già visto nei capitoli precedenti, è sempre più orientata allo sviluppo di reti come le Body Area Network (BAN) [8] e le Wireless Sensors and Actuators Network (WSAN) [23], e di applicazioni IoT orientate agli smart environment, nello specifico all'health monitoring e all'home assistant. L'implementazione di sistemi di questo tipo richiede numerose fasi di test e simulazioni in cui i dispositivi coinvolti devono scambiarsi grandi quantità di dati. Questi ultimi sono indispensabili per testare sia le applicazioni a bordo di ogni nodo sia le applicazioni distribuite su più nodi e, in molti casi, sarebbe utile fa riferimento allo stesso insieme di dati durante tutte le simulazioni, al fine di operare nelle stesse identiche condizioni. Ciò permette di valutare con maggiore precisione ogni modifica apportata al sistema, dal momento che un qualsiasi miglioramento o peggioramento delle performance dipenderebbe soltanto da tali modifiche e non dalle variazioni dei dati.

Il modello o, per meglio dire, lo strumento di testing che si vuole proporre prevede l'utilizzo di un sensore virtuale al posto del suo corrispondente reale. Virtualizzare un sensore significa utilizzare i dati contenuti in un dataset anziché quelli registrati da un sensore reale.

Normalmente i dataset vengono utilizzati in ambito di intelligenza artificiale e offrono il vantaggio di poter confrontare più reti neurali addestrate a partire dal medesimo dataset, dal momento che i dati sono, ovviamente, gli stessi. La logica alla base dell'idea sopra descritta è praticamente la stessa e quindi qualsiasi applicazione può essere facilmente confrontata con una sua simile.

Un altro vantaggio molto importante offerto dai dataset è la grande varietà di dati che essi contengono. In alcuni casi, i test richiedono ambienti e circostanze molto particolari, di cui ovviamente non tutti possono disporre in base al proprio contesto di lavoro e, non di meno, alla propria disponibilità economica: alcuni test necessitano di risorse che richiedono ingenti investimenti. L'argomento trattato nel capitolo precedente e il dataset che è stato utilizzato è un ottimo esempio di quanto appena detto.

Nello scenario più semplice la virtualizzazione di un sensore può essere effettuata da un unico dispositivo. In questo caso, il dataset viene caricato in memoria e i dati vengono letti ed elaborati direttamente in locale. Chiaramente l'approccio appena descritto vale soltanto se il dispositivo dispone di sufficienti risorse di memoria tali da poter memorizzare un intero dataset. In uno scenario più complesso, troviamo invece due dispositivi di cui il primo si occupa di memorizzare il dataset e di inviarlo progressivamente al secondo, il quale elaborerà i dati ricevuti. Quest'ultimo approccio è parecchio indicato per dispositivi dalle limitate risorse computazionali quali ad esempio i nodi di una rete smart come quella proposta.

Il sistema in esame segue quest'ultimo approccio ed è composto da un dispositivo wearable, rappresentato dalla SensorTile descritta nel precedente capitolo, e da un dispositivo fisso, rappresentato da un RaspberryPi 3 B+. Ovviamente, in questa tesi verrà trattata soltanto la parte wearable, mentre la parte riguardante il dispositivo fisso è oggetto di

un'altra tesi. Anche in questo caso si farà uso del dataset visto nel precedente capitolo, dunque, ciò che vedremo sarà nello specifico la virtualizzazione di un accelerometro. Nel prossimo paragrafo vedremo tutti i dettagli dell'esperimento svolto, il cui obiettivo è quello di valutare le performance della virtualizzazione confrontandola con l'utilizzo del sensore reale.

4.1 Descrizione e test del sistema

L'idea che sta dietro questo esperimento è quella di sfruttare la virtualizzazione dell'accelerometro per testare l'algoritmo per il rilevamento delle cadute trattato nel precedente capitolo, ma tale approccio si presta perfettamente per qualsiasi altro tipo di applicazione. Dunque, lo scenario prevederebbe che il nodo fisso invii, uno alla volta, i test contenuti nel dataset al nodo wearable, il quale li processerà con l'algoritmo e, nel caso in cui venga rilevata una caduta, restituirà al nodo fisso un messaggio di allarme.

In particolare, ciò che ci interessa valutare sono le performance della comunicazione BLE tra nodo fisso e nodo wearable relativa alla virtualizzazione del sensore. In altri termini ci interessa sapere se la virtualizzazione introduce una serie di complicazioni che potrebbero penalizzare eccessivamente questo approccio, se confrontato con l'utilizzo di un sensore reale. D'altronde, trattandosi di una comunicazione wireless è ragionevole ipotizzare la presenza di ritardi di propagazione. Per valutare questo aspetto, sono stati svolti ripetuti test per misurare con maggiore precisione i tempi di invio e ricezione rispettivamente relativi al nodo fisso e al nodo wearable. Inoltre, è stata valutata anche la reattività di risposta del nodo wearable, calcolando il tempo che intercorre tra l'invio di un messaggio da parte del nodo fisso e la sua ricezione. A breve sarà illustrato

più dettagliatamente quanto appena detto, intanto iniziamo col descrivere la comunicazione BLE tra i due nodi.

Come abbiamo già visto nel capitolo precedente, lo scambio di messaggi tra due dispositivi BLE è basato sui concetti di servizio e caratteristica. Per l'esperimento in esame, è stato utilizzato come base di partenza il pacchetto software di esempio della SensorTile [42], che di base mette a disposizione alcuni servizi e caratteristiche custom. Usufruendo di uno di questi servizi, sono state aggiunte due caratteristiche che per semplicità, d'ora in avanti chiameremo VirtualSampleIn e VirtualSampleOut. Come è facile intuire, la prima sarà utilizzata per ricevere i campioni dal nodo fisso, mentre la seconda per inviarli ad esso. A tali caratteristiche sono stati assegnati i permessi di READ, WRITE, WRITE WITHOUT RESPONSE e NOTIFY, di cui gli ultimi due sono quelli sfruttati durante l'esperimento. Nello specifico, in fase di ricezione, il nodo fisso può scrivere nell'apposita caratteristica senza aspettare una risposta di conferma (acknowledgement) da parte del nodo wearable, il quale, grazie al permesso NOTIFY, leggerà i dati ogni volta che il valore della caratteristica cambierà. Analogamente in fase di invio, ogni volta che il nodo wearable scriverà nella caratteristica il messaggio di risposta, questo verrà notificato al nodo fisso.

Di seguito riportiamo una figura riassuntiva che illustra come i due dispositivi, rappresentanti nodo wearable e nodo fisso, si scambiano i messaggi attraverso le predisposte caratteristiche.

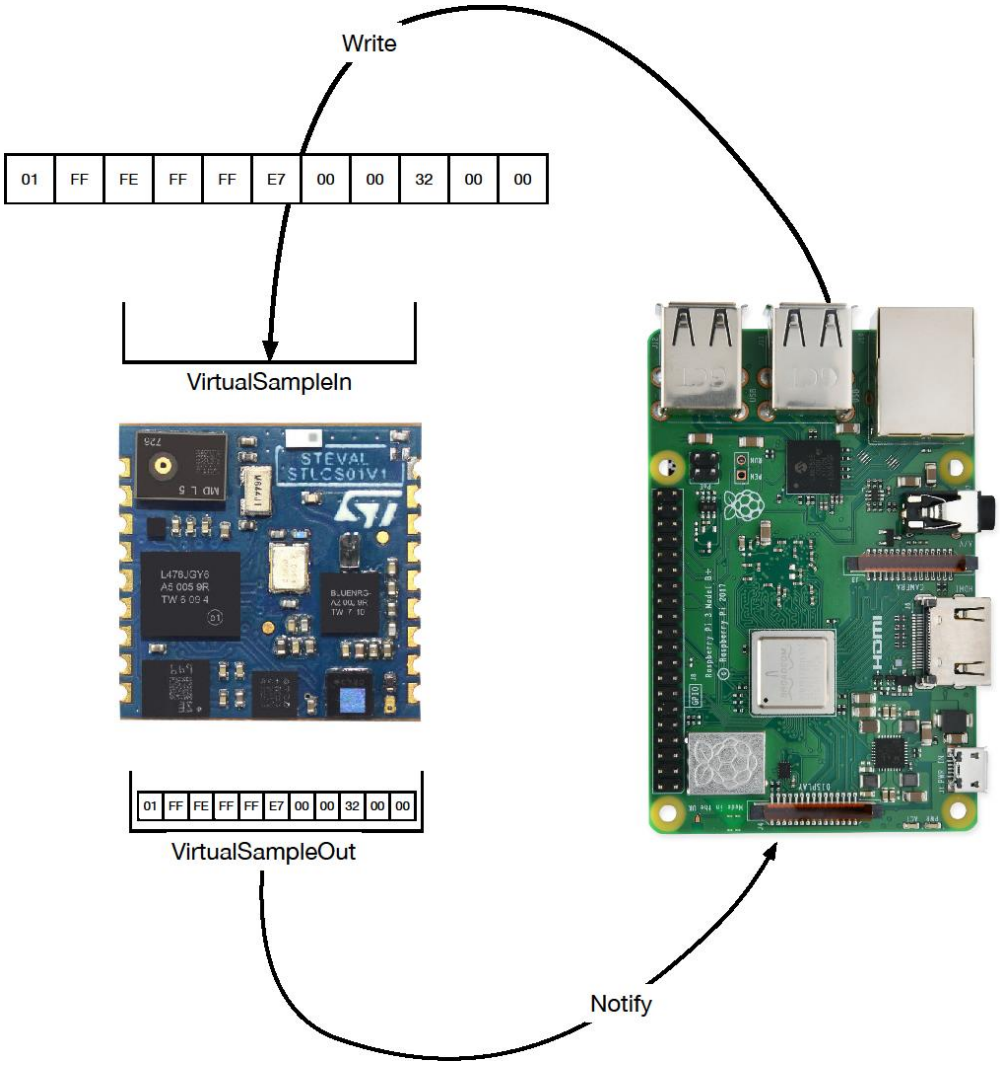

Figura 4.1 - Schema di comunicazione tra SensorTile e RaspberryPi

Nel prossimo paragrafo vedremo la struttura e il contenuto dei messaggi scambiati tra i due nodi.

4.1.1 Struttura dei messaggi

Innanzitutto, dobbiamo distinguere due tipi di messaggi, ognuno dei quali possiede una struttura ben precisa:

- Messaggio di configurazione: questo messaggio viene solitamente inviato dal nodo fisso subito dopo aver instaurato la connessione con il nodo wearable e contiene tutti i parametri necessari che quest'ultimo dovrà impostare ai fini della

virtualizzazione; questo messaggio può anche essere inviato durante la ricezione dei dati per modificare tali parametri.

- Messaggio dati: questo è il messaggio contenente i dati effettivi, cioè il payload costituito dai valori dell'accelerometro.

Di seguito si riporta uno schema raffigurante la struttura dei messaggi sopra descritti.

Figura 4.2 - Struttura del messaggio di configurazione

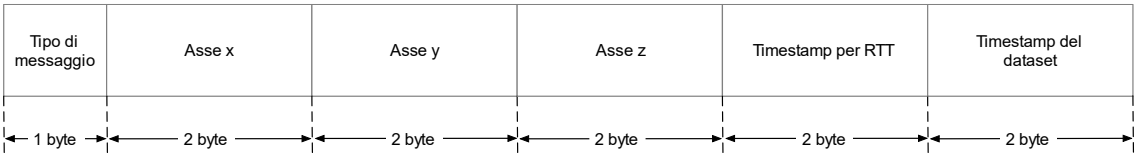

Figura 4.3 - Struttura del messaggio dati

Al fine di discriminare correttamente i due messaggi, viene riservato in entrambi un byte per identificare il tipo.

Il messaggio di configurazione, come si legge nello schema, ha una dimensione fissa di 8 bytes e contiene, oltre il tipo, i seguenti campi:

- Dimensione del payload (1 byte): indica quanti byte del messaggio dati sono riservati al payload;
- Numero di campioni (2 byte): il numero di campioni relativo al test che si sta inviando;
- Numero di ripetizioni (2 byte): il numero di ripetizioni del test.

Limitatamente all'esperimento in esame, la dimensione del payload è rimasta sempre la stessa poiché sono stati utilizzati 2 byte per ogni valore

dell'accelerometro, per un totale di 6 byte. Il secondo campo è necessario affinché il nodo wearable sappia il numero di campioni del test che sta per ricevere. L'ultimo campo serve invece a definire quanti test saranno inviati sequenzialmente durante la sessione dell'esperimento; il suo valore è chiaramente arbitrario e dipende dagli scopi che si desidera raggiungere.

Per l'esperimento in esame, poiché ogni file del dataset ha una lunghezza fissa di 3000 righe (ogni riga è rappresentata da una terna di campioni x, y e z) e poiché tutti i file hanno approssimativamente la stessa dimensione in memoria (circa 149 KB), si è scelto di utilizzare sempre lo stesso file durante tutta la sessione e di ripetere il test 100 volte al fine di ottenere risultati più accurati. D'altronde ciò che ci interessa è valutare i tempi/ritardi di invio e ricezione, i quali prescindono dal contenuto semantico del payload.

Il messaggio dati, invece, ha una struttura abbastanza differente e in questo caso ha una dimensione di 11 byte. Oltre ai campioni dell'accelerometro sono presenti due campi da 2 byte ciascuno: il "timestamp per RTT" e il "timestamp del dataset". Il primo è un timestamp che, come vedremo a breve, servirà per calcolare il tempo che intercorre tra l'invio di un messaggio e la sua ricezione da parte del nodo fisso. Il secondo è, invece, quel timestamp che solitamente accompagna ogni campione nei file dei dataset; questa informazione viene convenzionalmente riportata quando si fa il logging dei dati di un sensore e sta appunto ad indicare l'istante esatto in cui il relativo campione viene registrato. Questi timestamp potrebbero essere utili nel caso in cui si voglia lasciare al nodo wearable il compito della temporizzazione dei campioni, dal momento che è facilmente ricostruibile il ritardo che intercorre tra un campione e un altro, per di più in modo esatto. Nel caso in esame, il SisFall dataset non riporta questa informazione temporale, ma, per completezza, è stata comunque aggiunta come campo del messaggio dati.

Dopo aver visto la struttura dei messaggi, vediamo adesso come si è operato per valutare tutti gli aspetti temporali sopra citati.

4.1.2 Risultati sperimentali

Come già accennato, l'obiettivo di questo esperimento è quello di confrontare le misurazioni temporali ricavate dalla virtualizzazione con quelle ideali che si otterrebbero con l'utilizzo di un sensore reale. Osservando i file e le specifiche del dataset sappiamo che i campioni sono stati registrati alla frequenza di 200Hz e che ogni file, come già detto pocanzi, contiene 3000 terne di campioni x, y, z. Queste informazioni confermano che la durata di ogni test è di 15s, come riportato anche nell'articolo [34]. Tale valore è stato usato come riferimento per confrontare i risultati ottenuti dalla virtualizzazione.

Per virtualizzare correttamente il sensore in esame è indispensabile porsi nelle stesse condizioni sopra citate, ciò significa che è necessario rispettare la frequenza di campionamento. Nel caso in esame, questo compito viene assunto dal dispositivo fisso, il quale inserisce un ritardo tra l'invio di un campione e un altro; nello specifico questo ritardo corrisponde a 5 ms, ovvero l'inverso della frequenza di campionamento.

Inizialmente sono stati svolti alcuni semplici test in cui si misurava il tempo di ricezione di un singolo file (test) del dataset da parte del dispositivo wearable. Nello specifico, questa misurazione è stata effettuata avviando un timer alla ricezione del primo campione e arrestandolo alla ricezione dell'ultimo. Ripetendo più volte questo test, come era prevedibile, si è visto che i valori di tempo ottenuti erano leggermente diversi gli uni dagli altri, ovvero che ogni test non durava esattamente come il precedente, sebbene tutti fossero prossimi al valore ideale.

Per ottenere una misurazione più accurata e fare le dovute considerazioni, si è scelto di ripetere per cento volte consecutive l'operazione precedente, registrando sia la singola durata di ogni test, sia la durata complessiva di tutte le ripetizioni. Quest'ultima è risultata pari a 1509.908 s, ottenendo un valor medio di 15.1 s, cioè appena 0.1 s in più del valore ideale. Inoltre, avendo calcolato anche le durate di ogni singolo test, è stato possibile ricavare la varianza, corrispondente a $6.17 * 10^{-5}$. Questo valore dimostra che le variazioni tra la durata di un test e un altro sono davvero trascurabili e quindi si può affermare che il sistema in esame sia molto stabile.

Infine, è stato calcolato anche il Round Trip Time (RTT), ovvero il tempo che intercorre tra l'invio di un campione e la ricezione di quest'ultimo. Questa misurazione è stata effettuata dal nodo fisso, dal momento che ad esso spettava l'invio e la conseguente ricezione dei dati. Nello specifico ogni volta che un campione veniva inviato, il nodo fisso scriveva il timestamp relativo a quell'istante temporale nel campo "timestamp per RTT" e alla successiva ricezione di questo messaggio, calcolava la differenza con il nuovo timestamp. Questa operazione è stata ripetuta durante tutti i 100 test e alla fine si è ottenuto un valor medio di 26.64 ms.

Nel prossimo capitolo seguiranno tutte le considerazioni sui risultati ottenuti, valutando l'efficienza di questo strumento e le sue eventuali applicazioni in altre circostanze; infine, verranno discussi anche i possibili sviluppi futuri.

5 Conclusioni

Analizzando i risultati ottenuti si nota che lo strumento di virtualizzazione non introduce ritardi considerevoli confrontandolo con l'utilizzo del corrispondente sensore reale, infatti, come abbiamo visto, la durata media di ricezione dei campioni di un test è prossima al valore ideale. Inoltre, avendo ripetuto molte volte lo stesso test, si è potuta valutare la stabilità del sistema calcolando la varianza tra le durate di ogni singolo test. Il valore molto basso ottenuto indica che la durata di ogni test è pressoché la stessa per ogni ripetizione; in questo modo possiamo escludere la presenza di ritardi casuali significativi che potrebbero essere legati ai tempi di elaborazione o alla trasmissione e propagazione wireless dei dati.

Infine, il valore di RTT ottenuto indica una buona reattività di risposta del dispositivo wearable, ciò costituisce una buona base se si desidera implementare un'applicazione di media complessità, la quale introdurrebbe un ritardo di elaborazione più o meno significativo. In altri termini, partendo da un RTT molto basso, le performance della comunicazione sono meno vincolate dai tempi di elaborazione dell'applicazione, ovvero si beneficia di una tolleranza temporale maggiore.

Il sistema presentato in questa tesi costituisce ovviamente soltanto un esempio in cui si è deciso di coinvolgere un dispositivo fisso e uno wearable, ma ciò non toglie che esso possa essere esteso a scenari differenti in cui la virtualizzazione coinvolga anche nodi fissi e non soltanto wearable. Inoltre, si possono immaginare scenari ibridi in cui siano presenti sia sensori reali che virtuali ai fini di testare un'applicazione distribuita nella sua interezza. In questo caso, data la presenza di più sensori virtuali, sarebbe opportuno demandare la temporizzazione dei campioni ai nodi che li elaboreranno; chiariamo subito questo concetto. Nell'esperimento svolto era compito del

dispositivo fisso scandire i campioni ad una determinata frequenza e, come abbiamo visto, questo approccio è risultato efficiente. Tuttavia, nel caso di un sistema con molti sensori virtuali, il dispositivo fisso si ritroverebbe a gestire la temporizzazione di ciascuno di essi, nel peggiore dei casi anche contemporaneamente. In tal caso, seguire l'approccio adottato nell'esperimento visto in precedenza potrebbe risultare inefficiente, dunque, la soluzione sarebbe quella descritta pocanzi, ovvero lasciare il compito della temporizzazione al nodo che riceve i dati, il quale dovrebbe gestire una coda che viene progressivamente elaborata alla frequenza di campionamento specifica. Ciononostante, se da un lato questo approccio introduce dei benefici, dall'altro risulta un po' delicato, poiché bisogna prestare molta attenzione a non saturare la coda (ricordiamo che le risorse di questo genere di nodi sono il più delle volte molto limitate).

Infine, come già accennato nel terzo capitolo, questo tipo di approccio potrebbe anche essere integrato nello sviluppo di un sistema di smart environment basato sullo scambio di codice eseguibile [21] [22]. In uno scenario di questo tipo, si potrebbe godere del vantaggio di programmare interattivamente ciascun nodo, con la possibilità di modificare i parametri relativi alla virtualizzazione e l'applicazione che si desidera implementare nel nodo, il tutto senza necessità di ricompilare il codice sorgente. Inoltre, non si dovrebbe neanche rispettare una struttura dei messaggi ben definita, come quella descritta in precedenza, dal momento che verrebbero scambiati frammenti di codice eseguibile.

Bibliografia

[1] A. Procopio, «Alla scoperta dei dispositivi “wearable”: la tecnologia del futuro che migliorerà ogni aspetto della vita quotidiana,» 10 09 2017. [Online]. Available: https://www.ingegneriabiomedica.org/news/biotech-support/alla-scoperta-dei-dispositivi-wearable-la-tecnologia-del-futuro-migliorera-aspetto-della-vita-quotidiana/.

[2] «Steve Mann,» [Online]. Available: http://cyborganthropology.com/Steve_Mann.

[3] «Un tuffo nella storia delle Wearable Technology,» [Online]. Available: https://topweartech.altervista.org/un-tuffo-nella-storia/?doing_wp_cron=1579441711.5142259597778320312500.

[4] «Google Glass,» Wikipedia, [Online]. Available: https://it.wikipedia.org/wiki/Google_Glass.

[5] L. Longhitano, «I Google Glass sono tornati, ma solo per le imprese,» [Online]. Available: https://www.wired.it/gadget/accessori/2017/07/19/ritorno-google-glass-imprese/.

[6] «Pebble,» Wikipedia, [Online]. Available: https://it.wikipedia.org/wiki/Pebble_(orologio).

[7] «Body area network,» [Online]. Available: https://en.wikipedia.org/wiki/Body_area_network.

[8] D. Peri, «Body Area Networks and Healthcare,» in *Advances onto the Internet of Things: How Ontologies Make the Internet of Things Meaningful*, S. Gaglio e G. Lo Re, A cura di, Cham, Springer International Publishing, 2014, p. 301–310.

[9] I. Nonin Medical, «Unmatched Pulse Oximetry,» [Online]. Available: https://www.nonin.com/wp-content/uploads/2018/11/M-18-037-01-3150-with-BLE-Feature-Sheet-1.pdf.

[10] I. Nonin Medical, «PureSAT-Advantage,» 2008. [Online]. Available: https://www.nonin.com/wp-content/uploads/2018/09/PureSAT-Advantage-Brochure.pdf.

[11] Empatica, «Embrace,» [Online]. Available: https://www.empatica.com/embrace2/.

[12] Empatica, «Embrace Technical Specifications,» [Online]. Available: https://support.empatica.com/hc/en-us/articles/115000290003-Embrace-Technical-Specifications.

[13] Omron, «HeartGuide,» [Online]. Available: https://www.omron-healthcare.it/it/misuratori-di-pressione/heartguide.html.

[14] Philips, «Wearable biosensor - wireless remote sensing device,» [Online]. Available: https://www.usa.philips.com/healthcare/product/HC989803196871/wearable-biosensor-wireless-remote-sensing-device.

[15] D. Inc., «Presentazione del nuovo sistema CGM Dexcom G6. Sperimenta il potere di ciò che Dexcom G6 può fare per te.,» [Online]. Available: https://www.dexcom.com/it-IT.

[16] «Sistema distribuito,» Wikipedia, [Online]. Available: https://it.wikipedia.org/wiki/Sistema_distribuito.

[17] «Z-Wave,» Wikipedia, [Online]. Available: https://it.wikipedia.org/wiki/Z-Wave.

[18] «ZigBee,» Wikipedia, [Online]. Available: https://it.wikipedia.org/wiki/ZigBee.

[19] «Bluetooth Low Energy,» Wikipedia, [Online]. Available: https://it.wikipedia.org/wiki/Bluetooth_Low_Energy.

[20] A. De Paola, P. Ferraro, S. Gaglio, G. Lo Re, M. Morana, M. Ortolani e D. Peri, «An Ambient Intelligence System for Assisted Living,» in *2017 AEIT International Annual Conference (2017 AEIT)*, Cagliari, 2017.

[21] S. Gaglio, G. L. Re, G. Martorella e D. Peri, «DC4CD: A Platform for Distributed Computing on Constrained Devices,» *ACM Trans. Embed. Comput. Syst.,* vol. 17, p. 27:1–27:25, 12 2017.

[22] S. Gaglio, G. L. Re, G. Martorella e D. Peri, «A Lightweight Middleware Platform for Distributed Computing on Wireless Sensor Networks,» *Procedia Computer Science,* vol. 32, pp. 908-913, 2014.

[23] S. Gaglio, G. L. Re, G. Martorella e D. Peri, «A fast and interactive approach to application development on Wireless Sensor and Actuator Networks,» 2014.

[24] L. Delpiano, «Rapporto mondiale OMS sulla prevenzione delle cadute nell'anziano,» Torino, 2015.

[25] A. Bourke e G. ÓLaighin, «A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor,» *Medical engineering & physics,* vol. 30, pp. 84-90, 2 2008.

[26] A. Bourke, K. O'Donovan, J. Nelson e G. ÓLaighin, «Fall-detection through vertical velocity thresholding using a tri-axial accelerometer characterized using an optical motion-capture system,» *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference,* vol. 2008, pp. 2832-5, 2 2008.

[27] A. Lombardi, M. Ferri, G. Rescio, M. Grassi e P. Malcovati, «Wearable Wireless Accelerometer with Embedded Fall-Detection Logic for Multi-Sensor Ambient Assisted Living Applications,» 2009.

[28] R. Jafari, W. Li, R. Bajcsy, S. Glaser e S. Sastry, «Physical Activity Monitoring for Assisted Living at Home,» 2007.

[29] G. Vavoulas, M. Pediaditis, C. Chatzaki, E. Spanakis e M. Tsiknakis, «The MobiFall Dataset: Fall Detection and Classification with a Smartphone,» *International Journal of Monitoring and Surveillance Technologies Research,* vol. 2, pp. 44-56, 7 2016.

[30] C. Medrano, R. Igual, I. Plaza e M. Castro, «Detecting Falls as Novelties in Acceleration Patterns Acquired with Smartphones,» *PloS one,* vol. 9, p. e94811, 4 2014.

[31] K. Frank, M. J. Vera, P. Robertson e T. Pfeifer, «Bayesian Recognition of Motion Related Activities with Inertial Sensors,» 2010.

[32] T. Vilarinho, B. Farshchian, D. G. Bajer, O. H. Dahl, I. Egge, S. S. Hegdal, A. Lønes, J. N. Slettevold e S. M. Weggersen, «A Combined Smartphone and Smartwatch Fall Detection System,» in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015.

[33] «SISTEMIC: SisFall Dataset,» [Online]. Available: http://sistemic.udea.edu.co/en/investigacion/proyectos/english-falls.

[34] A. Sucerquia, J. Lopez e J. Vargas-Bonilla, «SisFall: A Fall and Movement Dataset,» *Sensors,* vol. 17, p. 198, 1 2017.

[35] STMicroelectronics, «STEVAL-STLKT01V1 - Sensortile development kit,» [Online]. Available: https://www.st.com/resource/en/data_brief/steval-stlkt01v1.pdf.

[36] STMicroelectronics, «LSM6DSM,» [Online]. Available: https://www.st.com/resource/en/datasheet/lsm6dsm.pdf.

[37] STMicroelectronics, «LSM303AGR,» [Online]. Available: https://www.st.com/resource/en/datasheet/lsm303agr.pdf.

[38] STMicroelectronics, «LPS22HB,» [Online]. Available: https://www.st.com/resource/en/datasheet/lps22hb.pdf.

[39] STMicroelectronics, «BlueNRG-MS,» [Online]. Available: https://www.st.com/resource/en/datasheet/bluenrg-ms.pdf.

[40] «The Hitchhikers Guide to iBeacon Hardware: A Comprehensive Report by Aislelabs,» Aislelabs, 3 ottobre 2014. [Online]. Available: https://www.aislelabs.com/reports/beacon-guide/.

[41] «GATT Characteristics | Bluetooth Technology Website,» [Online]. Available: https://www.bluetooth.com/specifications/gatt/characteristics/.

[42] «Embedded software samples for SensorTile, including sensor data streaming via USB and BLE, data logging on SD card, audio acquisition via USB and on SD card, and playback,» [Online]. Available: https://www.st.com/resource/en/data_brief/stsw-stlkt01.pdf.

[43] STMicroelectronics, «Getting started with the STEVAL-STLKT01V1 SensorTile integrated development platform,» [Online]. Available: https://www.st.com/resource/en/user_manual/dm00320099-getting-started-with-the-stevalstlkt01v1-sensortile-integrated-development-platform-stmicroelectronics.pdf.

[44] «GATT (Services and Characteristics),» [Online]. Available: https://www.oreilly.com/library/view/getting-started-with/9781491900550/ch04.html.

[45] «Bluetooth Low Energy protocol stack,» Nordic semiconductor, [Online]. Available: https://infocenter.nordicsemi.com/index.jsp?topic=%2Fsds_s132%2FSDS%2Fs1xx%2Fble_protocol_stack%2Fble_protocol_stack.html.