Politecnico di Torino

Tesi laurea magistrale

Progettazione e prototipazione di sistemi di sensing indossabili per training di riabilitazione clinici

---

Studente:
Fabio Sbarra

Relatori:
Giorgio De Pasquale
Andrea Acquaviva

10 luglio 2018

Anno Accademico 2017/2018

Indice

0.1 Abstract

Negli ultimi anni è cresciuto,in modo sempre più rilevante,l'interesse nello studio dei "movimenti della mano" soprattutto nell'ambito di:neuroscienze, ingegneria biomedica, robotica, interazione umano-computer (HCI) e intelligenza articiale (AI). Tuttavia, non trascurabile è il ramo dei sistemi "glove-based " per la riabilitazione medica e la valutazione siologica.
In questa tesi si descriverà,infatti, la progettazione e prototipazione di un sistema "glove-based " basato su una mini PCB, in grado di trasmettere via bluetooth i dati dei movimenti della mano di un paziente verso il pc controllato dal medico.In particolare, si tratta di pazienti che hanno subito gravi traumi quali ictus e/o infarti e che a causa di ciò devono ri-acquisire la corretta mobilità di una o entrambe le mani.
Lo scopo della tesi consiste nella realizzazione di un sistema sensorizzato completo da inserire in un apposito package e da montare,tramite magneti, su un supporto inserito sul "glove" che verrà indossato dal paziente.

0.2 Introduzione

Risulta evidente, nella vita di tutti i giorni, come l'uso delle mani ed in particolare il movimento delle dita sia fondamentale per lo svolgimento diun gran numero di "tasks" nella vita di tutti i giorni (proprio ora le mie dita si muovono sulla tastiera del mio pc per la scrittura di questa tesi). Pertanto è ampiamente giusticato il crescente sforzo, soprattutto nell'ultimo ventennio, di ricercatori, scienziati e tecnici del settore, nello sviluppo di terapie di riabilitazione motoria degli arti superiori.
La maggiorparte di queste è basata sul sistema guanto-microcontrollore pilotato via rmware da un'applicazione android sviluppata per smartphones e/o tablets.
La tecnologia "glove-based ha in generale innumerevoli vantaggi, rispetto alle "vecchie" terapie basate principalmente sull'esperienza dei sioterapisti e dei medici. Prima di tutto permette la "registrazione" contemporanea di più di un "movimento dinamico" del dito come : aerrare, pizzicare,premere etc...Secondo,fornisce uno strumento oggettivo per la valutazione di anche leggeri cambiamenti nelle capacità motorie della mano. Inne , tali guanti, permettono la valutazione della capacità dei pazienti nello lo svolgimento di compiti svolti nella vita di tutti i giorni.
La tesi proposta è suddivisa in diversi capitoli, in cui passo passo verrà descritto il sistema "basato-su-guanto" usato per il monitoraggio real-time di quantità come: accelerazione della mano da cui è possibile ricavare velocità e posizione; temperatura della mano; forza applicata dalle dita durante diversi tasks; di quanti gradi uno o più dita si ettono per svolgere un dato task e così via.
Il primo capitolo contiene una review generale dei principali sistemi glove-based svi-

luppati negli ultimi anni. Nella letteratura scientico-ingegnieristica un'ampio studio sui sistemi wearables basati su guanto è stato condotto da "Dipietro", si vedano [1] e [2] ma risalgono ai primi anni del duemila, pertanto è stato necessario fornire una lista più aggiornata e completa relativa a studi più recenti.

Capitolo 1

Capitolo 1 : Stato dell'arte degli "Wired-gloves"

Nella prima parte di questo capitolo è descritto il cosiddetto "State of the Art " delle tecnologie usate e delle relative applicazioni legate ai guanti sensorizzati; in pratica, viene fornita una panoramica generale della maggior parte dei dispositivi presenti in commercio, partendo dai più datati sino a quelli non ancora introdotti sul mercato. La second parte, invece, è più di carattere tecnico: vengono trattati i principali problemi in cui ci s'imbatte nel progettare un "glove" ed inoltre si realizzano confronti tra le diverse soluzioni descritte nella prima parte.

1.1 Stato dell' arte

In questa sezione si ricopre un'arco temporale di all'incirca 35 anni di tecnologie "glove-based " usate in ambito medico ed in particolare nel campo della riabilitazione motoria.
E' importante considerare che la mano umana è dotata di ben ventisette ossa con all'incirca 25 gradi di libertà di movimento (DoFs).Tali ossa, sono mosse grazie all'uso di diciassette muscoliinterni alla mano stessa e diciotto muscolidell'avambraccio. Pertanto, si tratta di un sistema articolato e complesso in grado di eseguire funzioni molto più complicate di ogni altro sistema esistente. Infatti, il "sistema-mano" è da sempre stata fonte d'ispirazione per ingegnieri e scienziati nella progettazione di protesi ed imparando dal modello di movimento umano, per realizzare braccia e mani robotiche.
La valutazione clinica delle funzionalità motorie della mano,richiede l'acquisizione di una grande quantità di dati diversi: la forza applicata dalle dite in azioni come il "pinching"(pizzicare, premere) e il "gripping"(aerrare); variazioni di temperatura della mano; variazioni nella sudorazione della mano, ma soprattutto il range di movi-

mento delle articolazioni delle dita, la cui ridotta mobilità impedisce lo svolgimento di compiti molto semplici come ad esempio aerrare un bicchiere.
Misure quantitative del range di movimento sono, di solito, realizzate usando goniometri meccanici e/o elettronici. Tuttavia è necessario un procedimento di misura molto lungo, che fornisce un'accuratezza limitata oltre ad una scarsa ripetibilità delle misure stesse anche nel caso siano svolte da terapisti specializzati e di grande esperienza. Sfruttando, invece, sistemi basati su guanto è possibile misurare, contemporaneamente, il range di movimento delle articolazioni di tutte le dita, riducendo di molto i tempi di misura ed incrementando la percentuale di ripetibilità degli esercizi svolti.
Da quanto detto sinora, in generale uno strumento "glove-based " può essere denito come un sistema composto da un'insieme si sensori per la misurazione di speciche quantità, un circuito elettronico per l'acquisizione ed il processing dei dati in uscita dai sensori ed una power supply opportunamente distribuita. Di solito i sensori sono cuciti su un supporto in tessuto (il guanto appunto) indossato dai pazienti per la misurazione dei dati relativi ai movimenti della mano.
Negli anni, un gran numero di guanti con designs dierenti è stato proposto: ciascuno con diversi sensori ed interfacce in grado di soddisfare applicazioni speciche e diverse fra loro.
Il primo sistema basato su guanto, che che si diuse ben presto in tutto il mondo è stato il "Data Glove" [3], messo in commercio negli Stati Uniti nel 1987. In particolare il Data Glove è considerato uno dei primi wired glove introdotti in commercio, ed è stato realizzato in collaborazione con Jaron Lanier e Young L.Harvill. Il primo, insieme con Zimmerman ha lavorato sulla realizzazione di sensori magnetici e ultrasonici in grado di eseguire il tracking della posizione della mano. Tali sensori sono stati inseriti sia sul Data glove che poi sul Power glove realizzato per la Nintendo nel 1989. Il secondo ha introdotto i sensori ottici usati per misurare il bending delle dita. Un totale di dieci bending sensors di cui cinque per misurare il grado di essione delle articolazioni MCP(Methacarpophalangeal) e gli altri cinque per le articolazioni PIP(Proximal-interphalangeal). La seguente immagine mostra le diverse articolazioni delle dita di una mano, per meglio comprendere quanto descritto nora:

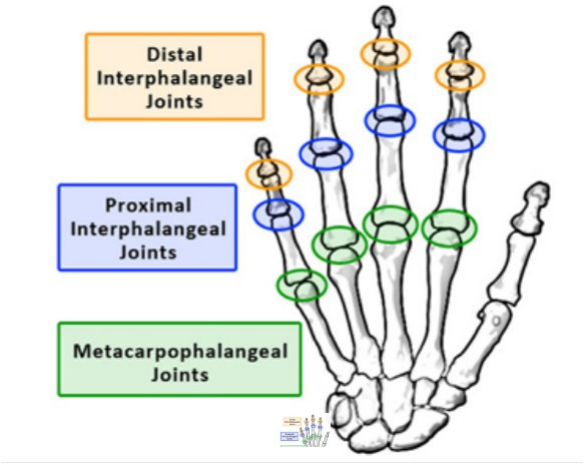


Figura 1.1: articolazioni mano

Nel 1997 lo Humanglove è stato sviluppato dalla compagnia Humanware [4], "spin-o" della Scuola Superiore Sant'Anna di Pisa (Italia). Lo scopo di tale guanto era misurare gli angoli di essione delle articolazioni delle dita attraverso 20 sensori basati sull'eetto Hall (dunque 4 per ciascun dito) ed inoltre i movimenti di abduzione/adduzione delle dita fra loro e del pollice da/verso le altre dita. Più o meno nello stesso periodo (1996) è stato sviluppato nell'Università di Shield (Inghilterra) il guanto SIGMA(Sheeld Instrumented Glove for Manual Assessment) con l'obiettivo, anche in questo caso, di valutazioni cliniche di tipo goniometrico dei movimenti della mano ([5] e [6]).
Esattamente dieci anni dopo, nel 2007, è arrivato sul mercato lo Shadow glove Monitor [8], basato su otto sensori resistivi per la misurazione degli angoli di "bending" delle dita. Questo nuovo tipo di guanto può funzionare in tre modalità, in ognuna delle quali vengono misurati i dati relativi ai movimenti della mano del paziente in modi diversi. La giusta modalità è selezionata attraverso il software con cui si controlla il guanto. I tre tipi di funzionamento sono:

- Sample-and-Send Mode, in cui i dati in uscita dai sensori vengono direttamente inviati ad un terminale (PC) e non sono immagazzinati nella memoria interna del sistema-guanto.
- Sample-and-Save Mode, in questo caso i dati sono salvati nella memoria interna del sistema-guanto.
- Sample-and-Dump Mode, una "mode" che mette insieme i due modi di funzionamento appena descritti. Infatti, permette l'acquisizione e trasmissione diretta dei dati dal glove verso il pc e contemporaneamente (se sono presenti dati in memoria) invia i dati dalla memoria interna verso il pc no a "svuotarla" completamente.

Nel 2009 è stato introdotto un sistema basato su guanto basato sull' harvesting dell'energia meccanica prodotta dalle oscillazioni della mano [56]. Tale tecnologia è applicata, in particolar modo, ai pazienti aetti da sindromi quali il Parkinson e l'Arthritis per ridurne il tremore delle mani. Ingegnieristicamente parlando, è come se fosse applicata una retroazione negativa al sistema mano; il rumore in questo caso è rappresentato dalle vibrazioni prodotte dai tremori delle mani; tale rumore viene misurato da un blocchetto realizzato con sensori piezoelettrici, che forniscono in uscita un segnale in tensione proporzionale all'energia meccanica d'ingresso. Condizionando opportunamente tale tensione si riescono a ridurre le oscillazioni delle mani dovute a tali malattie. Sempre nello stesso anno, è stato sviluppato da Gentner e Classen il WU glove, nell'università tedesca di Würzburg [10]. Si tratta di un guanto a basso costo basato su sensori di bending di tipo resistivo (14 in totale) che permettono inoltre la misurazione dei movimenti di adduzione/abduzione.Lo scopo è sempre quello della valutazione dei movimenti delle dita nell'ambito della riabilitazione neurosica.
Tuttavia occorre menzionare uno dei wired-glove che ha avuto maggior successo ed è tuttora in commercio: il Cyber Glove. La sua prima versione risale al 1990 ,[11], ed è stata sviluppata nel Virtual Space Exploration Laboratory del Centro di Ricerca Applicata dell'università di Stanford, nell'ambito del progetto "Talking Glove" per applicazioni di riconoscimento del linguaggio dei segni. E' stato, successivamente, messo in commercio dalla Virtual Technology Inc.per applicazioni come:realtà virtuale (VR), telerobotica e video games.Tale versione era basata su 18 sensori piezoelettrici per misurare i movimenti delle dita; nella seconda versione il Cyber Glove II si è passati a ben 22 sensori piezoelettrici. Nel 2009 è stato introdotto sul mercato il Cyber Glove III, dalla nuova compagnia Cyber Glove Systems LLC. Questo nuovo strumento può supportare no a 22 sensori di bending collocati in prossimità delle articolazioni DIP,MCP e PIP di ciascun dito, tra un dito e l'altro per misurare abduzione e adduzione, e alcuni anche sul dorso della mano.

Figura 1.2: Cyber Glove III

Nel 2010 è stato sviluppato e prodotto il Neuro-Assess Glove,[12], sempre nell'ambito della valutazione clinica delle funzionalità della mano e della riabilitazione.

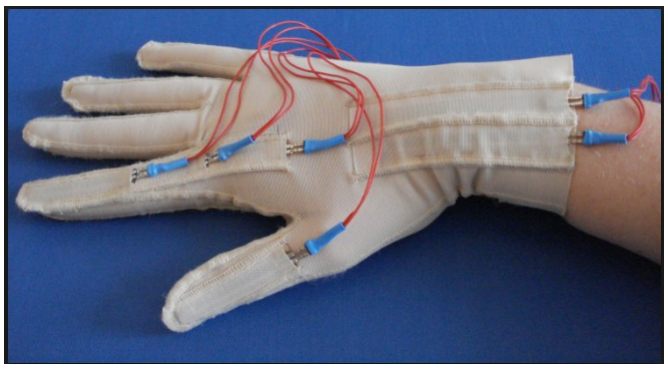


Figura 1.3: Neuro-Assess Glove

L'anno seguente (2011),[13], è stato introdotto sul mercato l'X-IST GloveQuesto guanto era disponibile in due versioni:la prima dotata di una combinazione di 5 sensori di bending più 5 sensori di pressione per misurare la forza applicata da ciascun dito collocati sulla punta di ciascun dito; la seconda, invece, da 15 sensori di bending più un accelerometro a due assi (x e y)I sensori di bending sono posizionati in prossimità delle tre articolazioni di ciascun dito (vedi gura 1.1) e si trovano su un guanto di cotone interno.All'esterno sono protetti da un'altro guanto realizzato in seta.
Un sistema-guanto più avanzato e completo è l'Atlas,[15], introdotto nel 2012Si tratta di un device costituito da due guanti per la riabilitazione, "da-casa", di pazienti colpiti da infarti o forti traumi, per migliorare le funzionalità degli arti superiori e in particolare delle mani.L' Atlas è composto da quattro blocchi principali: i due guanti, una IMU o inertial measurements unit, una base su cui appoggiare i guanti e un unità elettronica.In particolari i sensori di bending, montati sui guanti, sono basati su potenziometri (dunque di tipo resistivo).
Il 2013 porta ad un ulteriore svolta; infatti, è stato introdotto il Tyndall-UU Glove,[16], in cui l'elettronica per lo storage e l'analisi dei dati, per il controllo dei sensori ed i sensori stessi, sono stati collegati e saldati insieme su un unica exible-PCB. In questo modo è stata ottenuta una struttura in grado di adattarsi ai movimenti della mano e tuttavia mantenendo dimensioni ridotte, come mostrato nella gura seguente:

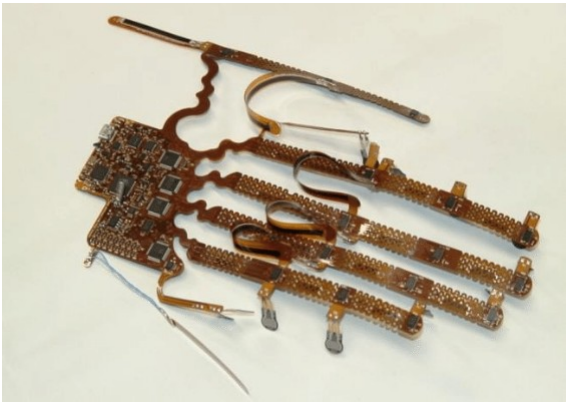


Figura 1.4: Tyndall-UU Glove

Il nome stesso del guanto si riferisce agli Istituti che hanno collaborato alla progettazione e allo sviluppo di tale dispositivo:Tyndall National Institute e l'Università di Ulster, entrambi in Irlanda. Lo sviluppo di questa nuova tipologia di guanto nacque per facilitare il processo di riabilitazione di pazienti aetti da Artrite Reumatoide (RA): malattia che attacca la membrana sinoviale delle articolazioni scheletriche riducendo, in particolar modo, il cosiddetto ROM (Range of motion) delle dita. E' stato, pertanto, sviluppato un guanto intelligente integrando sensori, processori e la tecnologia wireless per misurare empiricamente il ROM. Nello stesso anno, in India, è stato sviluppato un guanto,[17], dotato di sensori di forza sulla punta delle dita (FSR (Force sensing resistor)), per misurare la forza applicata dai pazienti nello svolgimento di precise attività motorie.
L'articolo scritto da Zhojie Ju e Honghai Liu del 2014, [19], è proposto un sistema multisensore in grado di valutare le traiettore percorse dalle dita e di estrarre segnali elettrici relativi alle forze applicate e ai muscoli usati durante l'esecuzione di una precisa attività motoria. Tale sistema è costituito da:un Cyber Glove (vedi [11]); da un FingerTPS, [10], ossia un dispositivo wireless per la misurazione della forza , basato su sensori di pressione; un sensore EMG (electromiography) per valutare lo stato di salute dei muscoli e delle cellule che li controllano, ovvero i motoneuroni.
Gli wired-gloves o Data gloves visti nora rappresentano un buon numero dei dispositivi di sensing motorio indossabili su una manoIn particolare, dieriscono tra loro per il modo in cui le informazioni cinematiche, dunque relative al movimento delle dita, vengono ottenute.I due metodi , più diusi, sono rappresentati dall'uso di sensori di bending resistivi e sensori di bending ottici (anche se spesso sono usati anche sensori piezoelettrici), questi ultimi sono più costosi ma permettono di ottenere una risoluzione più elevata (minore di 1 grado). Il problema di entrambi i metodi è il dicile piazzamento dei sensori stessi in

prossimità delle articolazioni delle dita. Per tale ragione, spesso, è necessario un processo di ri-calibrazione prima dell' utilizzo del sistema-guanto per ridurre l'errore di misura dovuto al non corretto allineamento del sensore con l'articolazione. Nel tentativo di eliminare tale problema, è stato sviluppato nel 2015 un nuovo tipo di data-glove, [20], basato su un sensore tattile realizzato in tessuto essibile e in grado di estendersi.In tal modo si ha un sensore con cui si possono ottenere aree di sensing diverse sulla stessa traccia in tessuto; inoltre può assumere più forme adattandosi ai movimenti della mandn questo caso è stato progettato un guanto dotato di 54 regioni sensibili alla pressione (in pratica dei sensori di pressione), come mostrato nella seguente gura:

Figura 1.5: wearable data-glove

Nello stesso periodo, è stato sviluppato un'altro "Glove module", [21], basato su sensori di pressione e angular sensors usati per misurare la rotazione di ogni articolazione di ciascun dito. Su ogni dito del guanto sono stati collocati, inoltre, dei vibration motors, ossia piccoli motori elettrici che forniscono precisi stimoli durante gli esercizi di riabilitazione; a ciascun motore è associato un LED che indica al paziente quale dito dovrà essere mosso (in totale 5 motori e 5 LEDs). guanto, in aggiunta, traccia i movimenti di traslazione e rotazione della mano sfruttando una MPU (Motion Processing Unit) costituita da un accelerometro e da un giroscopio. I dati raccolti vengono trasmessi in modo seriale ad un modulo HMI gestito da un software graco, che monitora i movimenti di ciascun dito e implementa specici esercizi riabilitativi.
Il 2016 ha visto apparire sul mercato il FuncAccess Glove, [23], che si basa principalmente sull'uso di sensori di bending e di forza di tipo resistivo.Tuttavia nello stesso anno è da considerare di maggior rilievo lo sviluppo di un'innovativa tipologia di sensing glove, [24].Questo guanto è fondato su un nuovo metodo per stimare le forze di contatto applicate dalle punte delle dita su oggetti deformabili.

In particolare è usata una struttura MARG basata su sensori magnetici, angolari e di gravità (Magneric Angular Rate Gravity) e appunto nessun sensore di forza.

Figura 1.6: Fun Access Glove

L'anno scorso (2017) si è tenuto a Dusseldorf (Germania), tra il 13 e il 16 Novembre, il "Wearable Technologies Show 2017 MEDICA,[29]; durante la quale sono state mostrate al pubblico europeo le principali applicazioni nell'ambito della riabilitazione medica del paziente.Tra le numerose novità, è stato riportato il "sistema-guanto" SEM Glove prodotto in Svezia dalla Bioservo Technologies, [25]. Si tratta di un guanto innovativo basato sulla tecnologia non invasiva SEM (Soft Extra Muscle). E' stato progettato per per persone con funzionalità della mano ridotte e/o per persone che svolgono lavori che necessitano di una presa "intensa ed ecace". Il SEM glove, pertanto, ha come scopo principale quello di aumentare il "grip" ed aggiungere forza e resistenza alla mano.

Figura 1.7: SEM glove

1.2 "Issues" nella progettazione di un sistema-guanto

In questa sezione, saranno analizzati gli aspetti concreti riguardanti il "sistema-guanto". Iniziando dalla scelta del materiale da usare per realizzare il guanto, no ad arrivare alla denizione dell'opportuna interfaccia graca. In ognuna delle seguenti sottosezioni verranno proposte le possibili soluzioni introdotte dai "scientists" menzionati nella parte iniziale di questa tesi.

1.2.1 Materiali con cui realizzare il guanto

Il materiale con cui è realizzato il guanto, rappresenta la base su cui i sensori, i li e la PCB saranno montati e collegati fra loro.
La maggiorparte dei guanti, menzionati nella prima parte, sono stati realizzati in Lycra. E' importatante ricordare che "Lycra" è uno dei più importanti marchi commerciali di questo prodotto; il materiale è lo Spandex (o elastam), ossia una bra sintetica di poliuretano molto usata per elasticizzare i tessuti. Tale bra è stata spesso scelta poichè garantisce un migliore contatto tra la pelle e l'oggetto aerrato, rispetto agli altri materiali. Tuttavia, col passare degli anni e con i miglioramenti delle tecnologie produttive, per favorire l'integrazione dei li di collegamento tra sensori e PCB all'interno del guanto, sono stati progettati e realizzati tessuti in grado di condurre l'elettricità: si parla dei cosiddetti "electro-textiles ",[30].
Il termine electro-textiles, spesso abbreviato con e-textiles, si riferisce a tessuti che mantengono le caratteristiche di essibilità, elasticità e lavabilità delle bre sintetiche, ma allo stesso tempo garantiscono una buona conduttività elettrica. In tal modo, i componenti e le interconnessioni sono intrinseche al tessuto e perciò meno visibili, riducendo il rischio di attorcigliarsi o strapparsi a contatto con oggetti esterni. Dunque, una corretta integrazione del "percorso conduttivo" (di solito son usati li di rame) nel tessuto (bre di poliestere) con cui è fabbricato il guanto, garantisce un rapido adattamento alle necessità computazionali e di "sensing" di una specica applicazione.
Gli e-textiles, dunque, permettono la progettazione di guanti, che rientrano nella più ampia classe degli "wearable computing devices", ossia indumenti in cui sono integrati sistemi elettronici atti a soddisfare applicazioni diverse.
Un terzo modo di realizzare il "sistema-guanto", consiste nello sfruttare una exible-PCB. Facendo riferimento al progetto di Tyndall,[16], il dorso della mano è ricoperto da un PCB essibile di 8 strati (spessore 0.45 mm) e avente dimensioni totali di 202.5mm x 124.9 mm.

Figura 1.8: essione del Tyndall glove

Si può notare nella gura 1.8, come la struttura della PCB essibile costituisca il guanto stesso.Lungo le dita sono stati adottati opportuni sensori di bending e di forza (di tipo SMD con footprint non inferiori agli 0402 pollici). La parte della PCB che realizza le "dita" è stata progettata con una particolare forma "serpeggiante" e allungata in modo da consentire essioni ripetute, senza arrivare a rottura. In tal modo la "PCB-guanto" può essere sottoposta ad ampie deformazioni fornendo elevate essibilità e robustezza, oltre che ad un buon livello di essibilità.
Un ultima soluzione è quella proposta dai ricercatori del "Neuroinformatics Group, Center of Excellence Cognitive Interaction Technology (CITEC), Bielefeld University, D-33619 Bielefeld, Germany", [20], che hanno progettato un tessuto innovativo, più essibile ed elastico rispetto a quello dei guanti precedenti, e allo stesso tempo in grado di funzionare come sensore di pressione. Dunque, si tratta di un "sistema-guanto" in cui i rigidi arrays di sensori tattili sono stati sostituiti da una bra, che emula la pelle umana, dotata di numerose "celle tattili" integrate in un singolo pezzo di tessuto (54 celle in totale).
Tale "guanto-sensore permette la misurazione di pressioni nell'intervallo tra 1kPa e 500kPa, che ricopre il comune range di forza, applicate dalla mano, nello svolgimento dei compiti di tutti i giorni. Inoltre, grazie alla sua struttura straticata, il sensore è molto robusto e può resistere a forze d'intensità superiore a quelle che potrebbero essere raggiunte da una mano umana senza subire danni.

1.2.2 Scelta dei sensori

I sensori rappresentano i "devices" che concretamente misurano le quantità relative al movimento della mano.La prima scelta fondamentale che un designer e/o produttore dovrebbe prendere, consiste nella selezione del tipo dei sensori da impiegare, in base all'applicazione scelta; inoltre risulta importante determinarne il numero e la rispettiva collocazione nel "sistema-guanto" considerato.Un approccio puramente teorico per arontare il problema di scegliere il numero più appropriato

e le relative locazioni dei sensori, fu proposto dai due "scientists" D.J.Sturman e Zeltzer, [31]. Tuttora in molti si basano o prendono spunto da tale metodo. Nell'ambito della riabilitazione clinica della mano, tra le numerose grandezze da stimare, sicuramente la quantità più importante è l'angolo di essione di ciascuna articolazione di ogni dito. I sensori di bending sono impiegati per tale scopdn particolare si distinguono quattro tipologie principali:

- ® sensori ricoperti da un lm di materiale piezolelettrico: il layer piezoelettrico funziona molto bene con i guanti basati sugli e-textiles, ed inoltre sono spesso usati anche per applicazioni di wearable computing in quanto orono bassi consumi di potenza e un'ampia scelta di "form factors".I materiali piezoelettrici rispondono quasi ad ogni tipo di stimolo sico con diverse intensità. Occorre evidenziare poi, che per ottenere una tensione più alta all'uscita del sensore, solitamente si applica un processo di laminazione del lm piezoelettrico.

- ® sensori basati su bra ottica. Le loro caratteristiche principali sono : peso ridotto, essibilità e immunità alle interferenze elettromagnetiche (EMI), elevate sensibilità e reliability. Di solito, questo tipo di sensori sono usati in applicazioni di HRI o Human-Robot Interaction.
  In pratica, vengono realizzate delle guide d'onda 2D che si collocano sulle dita del "sistema-guanto". Ad un'estremità della bra si trova un LED, mentre all'altra estremità un fotorivelatore (fotoresistore o fotodiodo); quando la bra si ette, in seguito al movimento delle dita, sempre meno luce, trasmessa dal LED, raggiunge il fotosensore. Dunque maggiore è il bending minore sarà la luce ricevuta. Di conseguenza il segnale elettrico misurato in uscita al sensore ottico avrà un'intensità proporzionale all'angolo di bending delle dita.

- ® sensori di tipo resistivo.Presentano una struttura basata sull'inchiostro di carbonio, la cui resistenza varia a seconda del livello di bending del sensore stesso.Si tratta di dispositivi poco costosi e di lunghezza variabile, che pertanto permettono un facile adattamento alle dimensioni del guanto progettato. I sensori di bending, di tipo resistivo, più usati sono quelli in poliestere laminato prodotti dalla Flexpoint, [35] (vedi immagine seguente).

Figura 1.9: sensore Flexpoint

⑧ sensori ad eetto Hall. Si tratta di trasduttori la cui tensione d'uscita varia in risposta al campo magnetico applicato.Sono spesso usati come sensori di prossimità, posizione, velocità e corrente.
L'eetto Hall può essere spiegato brevemente, analizzando la seguente gura, [36]:

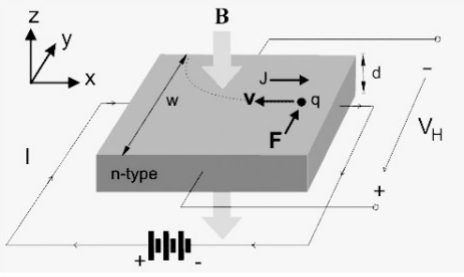


Figura 2: Schema dell'effetto Hall. Un semiconduttore di tipo n con una corrente I applicata e un campo magnetico B perpendicolare alla superficie. La differenza di potenziale V_H generata è noto come effetto Hall.

Figura 1.10: Eetto Hall

Si applica una tensione continua, una batteria ad esempio, in modo da generare una corrente I costante che scorre nel semiconduttore. Dopodichè, si applica un campo magnetico $\bar{B}$ perpendicolare alla supercie del conduttore scelto. In tali condizioni, gli elettroni (q) sono sottoposti sia all'eetto della forza elettrica $F_e = q * \bar{E}$, sia all'eetto della forza di Lorentz $F_l = q * (\bar{v} x \bar{B})$. I due contributi portano gli elettroni a percorrere una traiettoria curva, sino ad accumularsi da un lato del semiconduttore lasciando, di conseguenza, il

lato opposto pieno di cariche positive. Pertanto, si genera una tensione d'uscita nota come tensione di Hall(V_h). Ciò rappresenta proprio l'eetto Hall, ossia la generazione di una tensione attraverso un conduttore percorso da corrente e sottoposto ad un campo magnetico.
Lo Humanglove, [4], sfrutta appunto sensori di bending basati sull'eetto Hall.

In parecchi esercizi riabilitativi è necessario stimare anche i movimenti di adduzione delle dita verso il palmo della mano e quello opposto di abduzione, dunque delle dita che si allontanano dal centro della mano. Per tale scopo, sono appunto usati i sensori precedentemente descritti: sensori ad eetto Hall nello Humanglove, [4]; sensori ottici nel 5DT data glove, [37]; sensori resistivi nel WU Glove, [10].
Un'altra quantità fondamentale, che dev'essere stimata è la forza esercitata dalla mano su un'oggetto o da un dito su un'altro dito, durante lo svolgimento di uno specico training. Infatti, la forza esercitata può essere misurata sulla punta delle dita o sul palmo della mano attraverso sensori di diverso tipo. In letteratura sono presenti i seguenti:

- FSR (Force Sensing Resistors): Si tratta di semplici sensori passivi, basati sulla varizione della loro resistenza in risposta alla forza applicata: al crescere dell'intensità della forza, la resistanza del sensore diminuisce. I più noti, sono prodotti dalla InterlinK Electronics, [39], e dalla Tekscan, [40].

- Sensori di pressione piezoelettrici: Sono sensori che sfruttano l'eetto piezoelettrico, per cui si genera una dierenza di potenziale in risposta all'applicazione di una pressione meccanica. Nel caso del "sistema-guanto", si usano sensori realizzati con un tessuto piezoresistivo elastico costituito per il 72% da nylon e per il 28 % da spandex, prodotti dalla Eeonyx, [42].

- Sensori di pressione capacitivi: I più noti sono quelli usati nel sistema FingerTPS, [18]. Tale sistema è costituito da sei sensori capacitivi per ciascuna mano, connessi via wireless al pc. I sensori permettono la trasmissione delle informazioni ad un data rate massimo di 40 Hz e presentano un full-scale range di 10-50 lbs (4.35 - 22.7 Kg) ed una sensitivity di 0.01 lbs (0.45 Kg).

- Sensori EMG (Electromyography), usati nella misurazione dei segnali prodotti dalla contrazione dei muscoli nell'avambraccio, per stimare la forza applicata dalle dita, [11].

- Sistema di sensori MARG (Magnetic Angular Rate Gravity), [24] fornisce un modello cinematico dei movimenti della mano, combinando snsori magnetici,

sensori per la stima dell'angular rate e sensori gravimetrici (particolari accelerometri per misurare l'accelerazione di gravità).
Questo metodo è usato soltanto per stimare la forza applicata dalla mano su oggetti deformabili. Il MARG è installato sulle falangi e sul palmo della mano e permette di stimare la posizione delle punte delle dita rispetto al palmo. In pratica si sfruttano accelerometro e giroscopio per rilevare i cosiddetti "eventi di contatto": quando la punta di un dito tocca un'oggetto c'è una grande variazione di velocità in un breve intervallo di tempo (accelerazione).Una volta stimata la posizione delle punte delle dita in prossimità del punto di contatto con l'oggetto, è possibile determinare la forza applicata dalle dita all'oggetto attraverso la stiness matrix, [43].

Figura 1.11: FSR

La posizione della mano e la sua orientazione nello spazio, sono anch'esse di rilievo, per stimare i progressi riabilitativi del paziente. Durante i diversi trainings clinici, infatti, la mano del paziente percorre determinate traiettorie, che dopo esser state comparate con quelle compiute dall'arto di una persona sana, forniscono informazioni sulla stabilità e sulle posizioni spaziali.
I seguenti metodi sono stati adottati per valutare la posizione e l'orientazione nello spazio della mano:

- Metodo I : viene usata una MPU (Motion Processing Unit), [21]E' una "unit" costituita dalla combinazione di un'accelerometro 3D e di un giroscopio 3D, che permettono il tracking della posizione spaziale della mano; in aggiunta vi è un processore digitale per il controllo di questo sistema a 6 assi e per l'elaborazione dei dati ottenuti.Nel "sistema-guanto" citato è sfruttato un dispositivo MEMS a sei assi prodotto dalla TDK, InvenSense, [44], che permette di misurare l'accelerazione lineare e angolare.
  Simile a questo, è l'IMU (Inertial Measuring Unit) usata nel guanto [15]. L'unità inerziale in esame è montata sul dorso del guanto, in modo da poter stimare l'orientazione della mano rispetto ai 3 assi di un piano cartesiano e

quindi i cosiddetti movimenti di: rollio (Roll) rispetto all'asse x, beccheggio (pitch) rispetto all'asse y ed imbardata (Yaw) rispetto all'asse z.

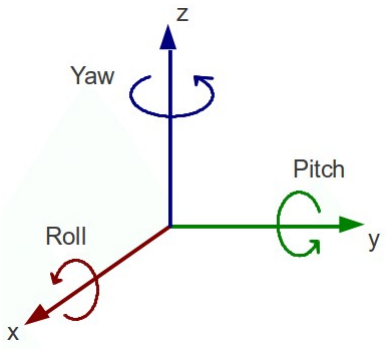


Figura 1.12: Roll, Pitch e Yaw

Perciò la IMU è composta da un sistema con 9 gradi di libertà: accelerometro a tre assi, giroscopio 3D e magnetometro triassiale. Inoltre è usato un microcontrollore Arduino per scrivere l'algoritmo di controllo di tali dispositivi.
Il sistema è dotato, anche, di un supporto su cui appoggiare mani ed avambracci, per standardizzare la posizione di partenza all'inizio di ogni esercizio riabilitativo; garantendo le stesse condizioni di misura iniziali per l'IMU.

- Metodo II, [16]: uso di accelerometri e sensori di bending in combinazione con il ltro di Kalman, [45].
  Quest'ultimo è denito come un'algoritmo di stima ottimale, che permette di determinare il parametro o la grandezza desiderata. Le applicazioni più comuni per cui è usato inclusono sistemi di guida e navigazione, sistemi di visione articiale (Computer Vision), [47], e l'analisi dei segnali (Signal Processing). Tale algoritmo fu introdotto da Rudolf E. Kalman; la prima applicazione importante del Kalman lter fu nel progetto Apollo, [48], per stimare la traiettoria della navicella spaziale verso la Luna e per il seguente ritorno sulla Terra.
  In particolare il Kalman lter è usato quando: le variabili d'interesse possono essere misurate soltanto indirettamente; le misurazioni sono ottenute con la combinazione di più sensori, per risurre l'eetto del rumore sulla misura stessa.
  Il guanto consiserato,[16], integra 16 accelerometri tri-assiali, uno per ciascuna falnge di ogni dito (3 falangi per 5 dita = 15 sensori) più uno sul palmo della mano. Tali accelerometri in combinazione con sensori di bending

sono impiegati , applicando l'algoritmo di Kalman, per stimare gli angoli di essione delle dita. Gli accelerometri, inoltre forniscono informazioni utili sui movimenti di rotazione e "shaking" della mano.

- Metodo III : si basa sull'utilizzo di una IMMS (Inertial and Magnetic Measurement System), che sfrutta l'algoritmo di Kalman.E' possibile dimostrare, infatti, che i sensori inerziali e magnetici sono piuttosto accurati nello stimare l'orientazione spaziale delle parti mobili del corpo umano, senza l'uso di videocamre esterne.
  Nel 2014 è stato introdotto un guanto, in ambito riabilitativo, basato sul sistema IMMS, [49]. Tale device impiega ltri di Kalman "extended " (EKF), [50], per stimare le traiettorie ottimali di mano e dita e poter denire, di conseguenza, un'accurata cinematica 3D della mano.

1.2.3 Posizionamento e disposizione dei sensori

Una volta scelta la tipologia dei sensori, la denizione del loro numero e della loro collocazione assume un ruolo fondamentale nella strategia di progetto del "sistema-guanto". Talvolta, infatti, piccole dierenze nella posizione spaziale dei sensori, può portare a grosse variazioni nel comportamento dell'intero sistema. Di seguito sono state proposte alcune delle possibili soluzioni presenti in letteratura.
I guanti [12], [10], [15], [23], sono dotati di particolari tasche in cui i sensori sono inseriti, in prossimità delle articolazioni delle dita. Spesso, la parte del sensore a cui saranno collegati i li d'interconnessione, è ssata al tessuto del guanto con del particolare nastro medico, pre evitare spostamenti causati dal movimento delle dita. Nel [12], inoltre, per evitare il contatto tra i diversi sensori piazzati sulle dita, sono stati cuciti tre strati di Spantex sul tessuto del guanto, formando così tasche separate in cui i sensori sono inseriti.
Nel [15], i sensori di bending sono stati collocati in apposite maniche in corrispondenza dell'indice, del medio, dell'anulare e del pollice. Tali maniche permettono ai sensori di scorrere sulle dita per adattarsi alle variazioni di lunghezza dovute ai movimenti di essione delle dita.Durante le estensioni, invece, le maniche impediscono che tali sensori si deformino verso l'alto, allontanandosi dalle dita.
Lo Shadow Glove Monitor, [8], invece, presenta una struttura totalmente dierente, in quanto è dotato di tasche in Lycra per ciascuna articolazione che dev'essere monitorata. Ciascuna tasca contiene un sensore di bending racchiuso in una sottile guaina di plastica. Ogni tasca è collegata alla parte dorsale del dito mediante

nastro adesivo di tipo medico, ad elevata tenuta. Da sottolineare che il nastro è applicato in modo da non ricoprire le articolazioni, garantendo libertà di movimento alle dita.
Altri tipi di guanto come il [17], non sono dotati di tasche. Un caso particolare è rappresentato dal [20]. Come detto in precedenza, infatti, il sensore di pressione è il materiale stesso con cui è realizzato il guanto; pertanto sull'intera supercie della mano sono state collocate numerose celle tattili perfettamente integrate nel tessuto del guanto stesso.
Il sistema-guanto descritto nell'articolo [49], è stato realizzato con stringhe di exible-PCB montate sul dorso della mano e in corrispondenza delle dita. Ciascuna stringa dispone di tre chip integrati ST LSM330DLC, [52], che contengono una coppia di accelerometro-giroscopio tri-assiali. In aggiunta un magnetometro 3D Honeywell HMC5983, [53], è montato sulla punta di ciascun dito e sul dorso della mano. I dati vengono campionati ed inviati tramite USB al computer impiegando un microcontrollore Atmel XMEGA.

Figura 1.13: Glove, exible-PCB, [49]

1.2.4 Tests di validazione

Dopo aver denito il numero e la collocazione dei sensori, occorre testare le loro prestazioni. Raramente, infatti, basando il proprio design su sistemi-guanto precedenti, sono fornite informazioni complete riguardo alle capacità di misura dell'intero sistema (elettronica + sensori) in termini di ripetibilità e precisione. Ciò è dovuto, soprattutto, alla mancanza di una procedura standard che denisca tests di validazione ecaci.
In questa sezione, infatti, sono stati riportati tests realizzati solo su sensori singoli ed in particolare su sensori piezoelettrici e di forza.
Negli articoli [54] e [55], viene citato il cosiddetto shaking test. In breve, si tratta di una procedura basata sull'uso di uno shaker elettrodinamico, [57], che ciclicamente applica un carico su un cantiliver piezoelettrico. Lo scopo di tale ricerca consiste nell'identicare il carico resistivo ottimale che massimizza la potenza elettrica

d'uscita generata. Inoltre, sempre nell'articolo [54] è stato proposto un secondo esperimento per esaminare l'energia liberata dall'impatto di piccole masse lanciate da un'altezza di 7 cm su un materiale piezoelettrico ssato ad una struttura rigida. L'articolo [59] , invece, è focalizzato sull'analisi del comportamento dei sensori di bending usati in un sistema-guanto per applicazioni neuroriabilitative. Viene sottolineata l'importanza del posizionamento di tali sensori, quando la essione non è applicata in modo omogeneo, come nel caso appunto della essione/estensione delle dita della mano. Infatti, gli esperimento svolti mostrano che uno spostamento longitudinale, anche di soli pochi millimetri, rispetto all'articolazione del dito, provoca variazioni nella curva tensione-angolo di bending del sensore considerato. Ciò conduce ad un problema non trascurabile: scarsa ripetibilità della misura il cui valore cambia a seconda della posizione iniziale del dito. Tale studio, pertato, ha lo scopo di determinare la posizione ottimale dei sensori di bending, che fornisca un'elevata risoluzione ed una buona ripetibilità di misura.
Una caratteristica comune a molti sensori di bending è il leggero "sliding" quando vengono mantenuti in una posizione ssata, bent position; un'altro termine d'inaccuratezza è dovuto al diverso valore di resistenza assunto dal sensore quando ritorna nella posizione iniziale (at), dopo essere stato piegato.
Negli articoli [12] e [8] sono stati eseguiti esperimenti su queste due tipologie d'errore, passando in rassegna diversi sensori in commercio. È stato ottenuto come risultato, che i sensori Flexpoint,[35], mostrano errori di misura più piccoli rispetto a quelli Abrams-Gentile e Spectra-Symbol, [60].
Nel [23] è proposto, invece, un test di valutazione della stabilità. Questo test è stato eseguito per determinare il drift (o deriva) del sensore, ossia di come varia nel tempo la sua uscita, in speciche condizioni di funzionamento. Il parametro drift è usato per valutare la variazione della resistenza nel tempo, sia per i sensori di bending a cui sono applicati angoli di essione diversi, sia per i sensori di forza sottoposti a carichi applicati diversi. Dunque si applicano carichi diversi e si valuta se la resistenza del sensore è la stessa in corrispondenza dello stesso carico oppure c'è un certo drift nel suo valore.
In generale, tuttavia, le prestazioni dei sensori di misura collocati nel sistema-guanto sono fortemente inuenzate dalla zero-position (o posizione di partenza) della mano e da come le dimensioni del guanto si adattano alla mano del paziente. Quest'ultimo è uno dei problemi più rilevanti nei sistemi-guanto a singola taglia dotati di sensori montati sul tessuto stesso.

1.2.5 Calibrazione dei sensori

Occorre ricordare che il sistema-guanto è usato su pazienti con mani di dimensioni diverse; di conseguenza, i sensori posti sul guanto potrebbero collocarsi in posizioni diverse per pazienti diversi.Ciò porta all'introduzione di un'oset di misura dierente per ogni utente, andando ad inuenzare le successive misurazioni.Per ridurre tali errori di misura è necessario eseguire un processo di calibrazione prima d'iniziare una seduta riabilitativa con nuovi pazienti. Purtroppo, il processo di calibrazione risulta spesso lungo in termini di tempo.Infatti, richiede al paziente di assumere determinate posizioni con le mani e di mantenerle per un certo intervallo di tempo.
In tal modo, si tenta di ridurre al minimo gli errori di guadagno e di oset dei sensori, per ottenere il miglior accoppiamento tra i movimenti della mano e le misurazioni dei sensori stessi.
Di seguito sono stati riportati alcuni dei processi di calibrazione più comunemente adottati per i guanti in analisi.
Il guanto [10], è dotato di sensori di bending con uscita in tensione direttamente proporzionale all'angolo di essione. Pertanto, per la calibrazione, sono stati necessari soltanto due valori di tensione corrispondenti rispettivamente a: condizione di riposo (o at hand position) e la posizione di massimo bending del sensore.Infatti, la calibrazione di ben 14 sensori è stata eseguita in soli dieci secondi.
Al contrario, nel sistema [8], la calibrazione è stata svolta in ben 8 minuti per paziente, poichè l'uscita dei sensori non era lineare.
Nell'articolo [15] sono descritti due processi di calibrazione diversi:uno relativo ai sensori di bending e l'altro per l'orientazione, nelle tre dimensioni spaziali, della mano. Il primo consiste nel posizionare le mani dei pazienti su speciche sagome, per far assumere alle dita gli angoli di essione corrispondenti:si parla appunte di misure statiche. In questo modo, infatti, le dita assumono determinate posizioni in modo da garantire la misurazione degli angoli di essione delle articolazioni MCP (Metacarpophalangeal) e PIP (ProximaInterphalangeal), di pollice, indice, medio e anulare usando un goniometro manuale.
Il secondo, è adottato per stabilire l'abilità del dispositivo nel misurare l'orientazione e la posizione nello spazio della mano.Dunque, dopo che il paziente ha eseguito con successo il protocollo relativo ai bending sensors, si passa a questo secondo protocollo sfruttando il sistema di misurazione inerziale IMU collocato sul guanto. Questa calibrazione è basata sul movimento del blocco polso-avambraccio-mano in sei diversi modi:il rollio (X), ossia il passaggio di tale blocco da una posizione prona ad una supina; il beccheggio (Y) realizzato tramite

la essione ed estensione del polso; l'imbardata (Z) , che consiste nella rotazione laterale del polso con conseguente movimento della mano verso l'ulna (ossia verso il dito mignolo, [61]) o dal lato opposto ossia verso il radio.
Il guanto descritto nel [23], è dotato anch'esso di sensori di bending con uscita non lineare. In questo caso, i valori di tensione d'uscita sono stati misurati in corrispondenza dei seguenti angoli di essione delle articolazioni delle dita: 0° (at position) , 20°, 40°, 60° e 90°.
L'analisi dei dati raccolti è stata svolta con Matlab e poichè la relazione tra la tensione d'uscita e gli angoli di bending non è lineare, è stata usata la funzione pchip per interpolare i valori ottenuti e generare una look-up table ad intervalli di 0,1.
Per quanto riguarda i sensori di forza, invece, sono stati calibrati applicando carichi di diverso valore: 10 g, 50 g, 100 g, 150 g, 200 g, 250 g, 300 g e 350 g. Misurando le tensioni d'uscita corrispondenti a tali carichi è stata ottenuta una relazione pressochè lineare per ogni sensore testato.
Il sistema-guanto descritto in [11], "CyberGlove", integra un sistema di calibrazione software, che permette la "calibrazione sul campo" del guanto da parte del dottore stesso, in base al paziente sottoposto alla riabilitazione. Tale software è noto come Virtual Hand ed è dotato di una calibrazione di base per il corretto controllo di questo "data glove". Tuttavia, per ridurre gli errori dei sensori è necessaria una calibrazione aggiuntiva, specica per ognuno di essi, come descritto per il [15].
Il "Cyber Glove III", invece, si basa su un sistema di calibrazione diverso. Infatti, si esegue la calibrazione dei sensori usando un "nto dito" (dummy nger), sul quale in modo automatico vengono testati i sensori di bending. Dopodichè, è stata realizzata una "lookup table", i cui valori rappresentano le relazioni tra le tensioni d'uscita dei sensori e gli angoli di essione corrispondenti. In tal modo, non è più necessario eseguire la calibrazione del sistema prima di ogni sessione riabilitativa; bensì sfruttare la tabella ottenuta ed in base ai valori di tensione letti in uscita, ricavare i corrispondenti angoli di essione delle dita. Si risparmia così una notevole quantità di tempo e si ottengono risultati di misura più precisi.
Il Tyndall/UU Glove, [16], in diretta concorrenza con il Cyber Glove III, è dotato di un sistema software per evitare la calibrazione all'inizio di ogni seduta. Infatti, i sensori di bending insieme con gli accelerometri permettono una misura accurata degli angoli di essione delle dita, nonchè della velocità e accelerazione della mano. Tutti i dati sono poi inviati al pc, che tramite il software li elabora e fornisce l'esatto posizionamento della mano nello spazio.

1.2.6 Analisi dei segnali (signal processing)

Terminato il processo di calibrazione, lo step successivo consiste nella realizzazione di un apposito circuito di condizionamento dei dati in uscita dai sensori. Nel caso del sistema-guanto occorre sfruttare un multiplexer, dispositivo che permette di analizzare in sequenza i segnali provenienti dai numerosi sensori presenti. Per ogni canale in ingresso al multiplexer occorrerà introdurre opportuni circuiti di condizionamento: più semplici nel caso le uscite dei trasduttori siano digitali, più complessi invece nel caso di uscite analogiche. Nel caso di sensori analogici, infatti, le elaborazioni più comuni sono le seguenti:

- convertire la grandezza elettrica in uscita dal trasduttore in una tensione;
- adattare il campo di variabilità del segnale a quello dei circuiti a valle (come ad esempio gli ADC) al ne di rendere massima la risoluzione del sistema di acquisizione;
- ridurre al minimo le alterazioni del segnale causate dai disturbi e dalle distorsioni;
- trasferire il massimo valore del segnale in tensione o in potenza;
- isolare i blocchi elettricamente per esigenze di sicurezza (ad esempio per applicazioni bio-mediche) attraverso opto-accoppiatori (o optoisolatori);
- ltrare il segnale, soprattutto se questo dev'essere campionato e convertito in digitale, inserendo un ltro anti-aliasing.
- compensare le non linearità del trasduttore;

Dopodichè a valle del multiplexer va inserito il sistema di campionamento dei dati costituito da: un Sample & Hold (S&H), un ADC e un microcontrollore, vedi gura seguente.

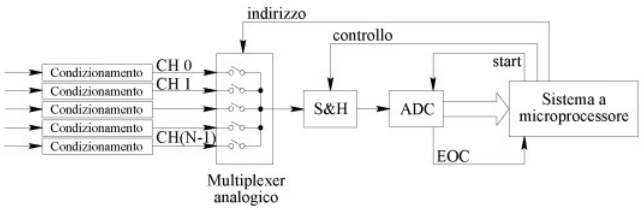


Fig.1.2 - *Sistema multicanale con multiplexer.*

Figura 1.14: Sistema di misura a più ingressi, [62]

Questa soluzione è economica, in quanto necessita di un solo S&H e di un solo ADC. Tuttavia, il primo svantaggio evidente è la riduzione della frequenza di campionamento consentita su ciascun segnale d'ingresso. Infatti, per un sistema d'acquisizione dati che presenta N canali d'ingresso, detta f_c la frequenza di campionamento massima alla quale può operare il sistema di campionamento e conversione (blocco ADC + S&H), la massima frequenza con cui potrà essere campionato il segnale i-esimo, fra gli N usati, risulta $f_i = \frac{f_c}{N}$. Questo risultato vale solo in prima approssimazione, trascurando il tempo di commutazione del multiplexer da un canale al successivo. Nella realtà, anche considerando i problemi legati allo "slew-rate" del Sample & Hold nel passaggio tra un canale e il successivo, si dovrà assumere per la frequenza di campionamento du ciascun canale, f_i inferiore a $\frac{f_c}{N}$.
Inoltre, nel sistema considerato in gura 1.14, i campioni dei diversi segnali d'ingresso risultano presi in sequenza e pertanto non è possibile disporre di campioni simultanei di più forme d'onda, come invece è richiesto per molte applicazioni. L'esigenza di eseguire un campionamento simultaneo su tutti i segnali in ingresso al sistema di acquisizione, può essere soddisfatta impiegando tanti campionatori S&H quanti sono i canali d'ingresso.

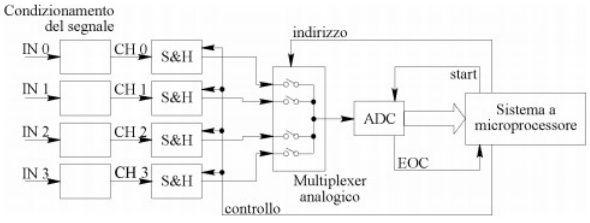


Fig.1.3 - *Sistema a campionamento simultaneo.*

Figura 1.15: Campionamento simultaneo, [62]

Dalla gura 1.15, si nota come tutti i S&H sono pilotati dallo stesso segnale di CLK ("controllo" nella gura). In tal modo gli N campioni prelevati dagli N segnali d'ingresso in un'acquisizione, risultano sincroni. Tali campioni, memorizzati nei S&H, sono poi convertiti in modo sequenziale dall'ADC.
Per ottenere, invece, un campionamento simultaneo su più canali e allo stesso tempo conservare la massima velocità di campionamento sul singolo canale, si va ad usare una coppia di S&H e ADC per ogni ingresso.

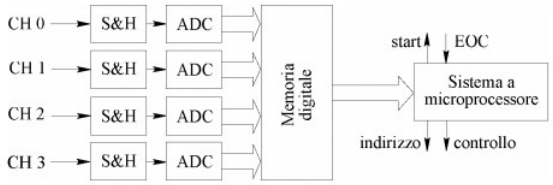


Fig.1.5 - *Sistema di acquisizione veloce.*

Figura 1.16: Sistema acquisizione veloce, [62]

Come mostrato in gura 1.16, il multiplexer viene sostituito da una memoria digitale gestita opportunamente dal microprocessore per la ricostruzione dei singoli segnali acquisiti.
Ad esempio nel sistema citato in [12], si sfrutta un semplice circuito di condizionamento per incrementare la risoluzione dei sensori di bending. I sensori usati sono da 2-inches e possono misurare angoli di essione nel range tra 0° e 120 gradi. Con l'opportuno circuito di condizionamento la curva tensione-bending può essere modicata per incrementare il range di misura d'ingresso e la risoluzione dei sensori usati.
Nell'articolo [10], l'idea è stata di posizionare un resistore in parallelo al sensore di bending e collegarlo all'ingresso di un amplicatore operazionale. Questo metodo è applicato per ciascun sensore di bending del sistema-guanto, in modo da ottenere una relazione lineare tra la resistenza del sensore e l'angolo di essione.
Il [23], invece, come circuito di condizionamento, propone un convertitore I-V. Nella gura seguente, è mostrato il circuito di condizionamento usato per un sensore di forza e un sensore di bending:

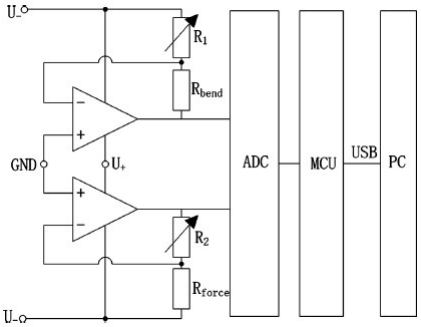


Figura 1.17: Circuito di condizionamento, [10]

La 1.17, mostra l'uso di un singolo ADC con il microcontrollore a valle. Dunque, la soluzione più economica, ma con lo svantaggio, come detto della riduzione della frequenza massima di campionamento per ciascun ingresso. In questo sistema di conversione si nota, inoltre, la mancanza del multiplexer (si hanno, appunto, 2 ingressi all'ADC); ciò poichè il MUX è integrato nel chip dell'ADC che racchiude

appunto: un convertitore analogico-digitale a 12 bits di tipo SAR (approssimazioni successive), un MUX a 16 canali, il S&H e l'opportuna logica di controllo, [63]. Oltre all'integrato appena menzionato, gli ADC più usati nei sistemi-guanto analizzati sono i seguenti:

- ADC a 8 canali, 125 KS/s della Analog Devices, [64];
- scheda di acquisizione dati USB-6008 della NI(National Instruments), con un 12-bits ADC a 8 canali, 10 KS/s, [65];
- 8-bits ADC 0808/0809 a 8 canali della National Semiconductor, [67];

I dati in uscita dall'ADC vanno in ingresso al microcontrollore. In generale i microcontrollori sono in grado di ricevere ed inviare dati sfruttando protocolli di comunicazione dierenti. I più usati sono: SPI (Serial Peripheral Interface), I^2C e UART (Universal Asynchronous Receiver-Transmitter).
Considerando le dimensioni ridotte delle PCB, che vengono montate sul sistema-guanto in ambito riabilitativo, la scelta del microcontrollore è fatta in base a due vincoli fondamentali: bassi consumi di potenza, trattandosi di un'applicazione wearable deve funzionare unplugged per lunghi intervalli di tempo; footprint ridotta per garantire ingombri ridotti in modo da non limitare i movimenti della mano del paziente. Pertanto gli MCUs più usati per queste applicazioni sono :

- famiglia HC12 della Freescale, [68];
- "PIC" della Microchip, [69];
- ATmega328 della Atmel, [70];
- MSP430 della Texas Instruments, [71];
- P89V51RD2 della Philips, [72];
- Xmega della Atmel, [73];
- Atmega1281 della Atmel, [74];

1.2.7 Interfaccia HMI

L'ultimo step della progettazione e prototipazione di un sistema-guanto in ambito medico, consiste nel realizzare un dispositivo d'interfaccia tra la macchina e l'utente, una HMI (Human Machine Interface). Questo modulo permette all'utente

nale (dottore, sioterapista o il paziente stesso) di disporre di un'interfaccia che comunica tutti i dati in maniera comprensibile. Tuttavia, è importante sottolineare che l'interfaccia HMI non è obbligatoria, in quanto i dati potrebbero essere gestiti direttamente sfruttando il software usato per programmare il Microcontrollore. Però, nel momento in cui si voglia realizzare una precisa interfaccia graca (GUI) con le relative animazioni, il modulo HMI è fortemente consigliato.
Tra i guanti analizzati, le soluzioni per realizzare il modulo HMI sono state le seguenti:

- Nel sistema descritto in [58], i dati in uscita dai sensori sono stati campionati ad una frequenza di 100 Hz usando l'ambiente LabView come software di controllo del circuito di condizionamento. Invece, nel [23] e nel [49] è stato usato uno script Matlab per ricevere, stampare a video e salvare i dati ricevuti dal guanto.
  Il sitema "Shadow Glove Monitor" [8], sfrutta un software scritto in NesC, ossia un'estensione del linguaggio di programmazione C, molto usato nelle applicazioni wearables basate sul sistema operativo TinyOS.
  In questi casi, dunque, l'interfaccia HMI è stata implementata in modo non graco, ma sfruttando software di elaborazione dati per pilotare opportunamente il microcontrollore.

- Un'approccio diverso, invece, consiste nello sviluppo di una vera e propria interfaccia graca (GUI) in grado di fornire animazioni 3D dei movimenti della mano del paziente, durante la seduta riabilitativa. Un primo esempio è rappresentato dal [75], in cui è stata realizzata una GUI (Graphic User Interface) programmando in C++ attraverso l'applicazione Windows API (Application Program Interface). Quindi, i movimenti della mano sono acquisiti dal wired-glove e sono stati, poi, riprodotti in tempo reale in un'ambiente di visualizzazione 3D usando il software Blender.
  Lo Humanglove [4], invece, è dotato di un software proprietario chiamato Graphical Virtual Hand. Tale programma è usato sia per la calibrazione del guanto sia per mostrare a video l'animazione di una mano che ricalca i movimenti svolti dal paziente.
  Il [21] e il [22] descrivono un sistema-guanto dotato di un'interfaccia graca simile a quella di un videogioco; questa applicazione graca è stata sviluppata per monitorare il movimento di ciascun dito e per implementare gli stessi esercizi riabilitativi che il paziente dovrà svolgere. La GUI è basata su una libreria graca della Microchip ed è stata sviluppata usando GDDX (Graphics Display Designer X). Sfruttando tale interfaccia "game-oriented" è possibile far esercitare il paziente in maniera meno noiosa e più divertente.

1.2.8 Power Management

In quest'ultima parte del primo capitolo, viene descritto il ruolo fondamentale della "PMU (Power Management Unit)", soprattutto per un'applicazione wireless come questa.
I sistemi-guanto di tipo wireless, infatti, sono caratterizzati da una PMU portatile contenuta all'interno del package stesso: nei casi meno recenti sono state usate batterie stilo "AA"; tuttora, invece, le più usate sono le coin cell batteries al Litio (Li) e le batterie "LiPo" (o ai polimeri di Litio), poichè garantiscono un maggior numero di cicli di funzionamento ed ingombri ridotti. Le batterie di tipo "coin", inoltre, possono essere ricaricabili.
Per esempio lo "Shadow Glove Monitor" [8] è dotato di due batterie di tipo "AA", che sono sucienti a garantire il suo funzionamento per ben 24 ore, con un'acquisizione dati continua.
Il [15] è dotato, invece, di batterie ricaricabili. E' stato realizzato un'apposito circuito per ricaricare la batteria a singola cella tramite la connessione USB tra il sistema-guanto e il pc.
Più recentemente è stata, inoltre, sviluppata una nuova tecnologia basata su un materiale composito in bra piezoelettrica, il PFC, [56], per realizzare l'harvesting dell'energia prodotta dai movimenti della mano stessa. In pratica, parte dell'energia sviluppata durante gli esercizi riabilitativi viene convertita in un segnale elettrico, che ricarica parzialmente la batteria.
Occorre sottolineare che la scelta del PFC non è stata casuale: infatti, i materiali polimerici sono più essibili, ma mostrano proprietà piezoelettriche ridotte. I materiali ceramici, invece, sono dotati di ottime capacità piezoelettriche, tuttavia sono rigidi e spesso sono prolati in blocchi troppo grandi, aggiungendo peso eccessivo al sistema. Il giusto compromesso, pertanto, per un'applicazione wearable di questo tipo è l'uso del PFC. Si tratta di una bra ceramica realizzata nella forma di lo a sezione circolare, con un diametro che varia tra 10μm e 250μm. E' usato un set di tali bre, che viene racchiuso in una matrice di resina epossidica. Dopodichè, tale struttura è racchiusa tra due strati di rame polimerico laminato, come mostrato nella gura seguente:

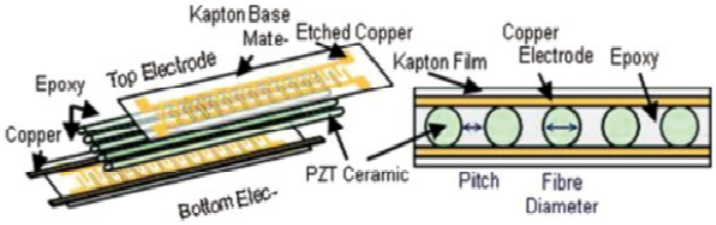


Figura 1.18: PFC, [56]

Un'importante caratteristica di questo materiale, è che permette di ottenere una tensione in uscita dai sensori, cinque volte maggiore rispetto ai lms polimerici. L'articolo [55], mostra un circuito basato su sensori piezoelettrici in grado di eseguire l'harvesting dell'energiaAll'uscita dei sensori, come circuito di condizionamento è inserito un ponte di diodi per "raddrizzare" i segnali. Dopodichè, a valle del circuito "bridge" si colloca un condensatore elettrolitico, detto condensatore di livellamento, che ha lo scopo di eliminare la pulsazione della corrente alternata, alzando il livello del valor medio della tensione.

Capitolo 2

Capitolo 2 : Analisi e debug della scheda prototipale

2.1 Speciche del dispositivo e training riabilitativi

In questo capitolo è stata descritta la struttura anatomica della mano, sottolineando l'importanza delle tecniche riabilitative per il recupero delle piene funzionalità motorie dell'arto.
In seguito, sono state presentate le tipologie di esercizi che il "sistema-guanto" dovrà svolgere ed inoltre sono state mostrate le diverse soluzioni sensorizzate che meglio soddisfano un determinato training di riabilitazione.

2.1.1 Anatomia della mano

La mano umana è dotata di 27 ossa che garantiscono all'incirca 25 gradi di libertà (DoFs). Tali ossa sono mosse da 17 muscoli interni alla mano stessa, e da 18 muscoli dell'avambraccio deniti per questo estrinseci.
Un sistema motorio così articolato è in grado di svolgere compiti molto complessi. Pertanto, è stata la fonte d'ispirazione per molti ingegnieri e studiosi del settore, per la progettazione di dispositivi robotici cosiddetti "human-like" e per la realizzazione di protesi per gli arti superiori.

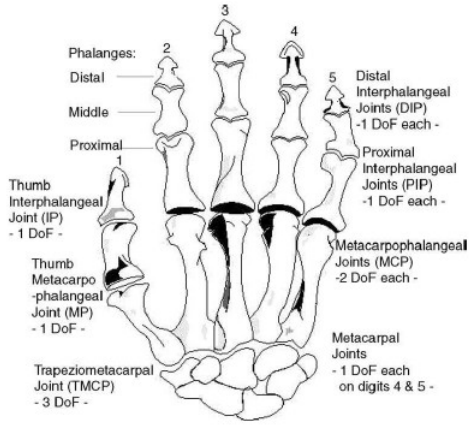


Figura 2.1: articolazioni della mano

La gura 2.1, mostra le principali articolazioni della mano e i diversi gradi di libertà tipicamente usati per descriverne i movimenti.
I gradi di libertà delle diverse articolazioni sono i seguenti:

- DIP (distal interphalangeal) e PIP (proximalinterphalangeal) di ciascun dito sono dotate di un solo grado di libertà, che consente loro l'esecuzione del movimento di "essione/estensione";
- MCP(metacarpophalangeal) dispone, invece, di due gradi di libertà infatti, oltre alla "essione/estensione" può svolgere un movimento di "abduzione/adduzione";
- TMCP(trapeziometacarpal) si trova solo nel pollice e gli permette di ruotare per entrare in contatto con le altre quattro dita;

E' stato provato e dimostrato, [2], che le funzioni motorie di mano e avambraccio si riducono di molto con l'avanzare dell'età a causa diriduzione della coordinazione motoria e della destrezza manuale, in aggiunta ad una debole forza nella presa degli oggetti. Spesso, tuttavia, nei soggetti anziani, la scarsa abilità manuale è causata da menomazioni dovute a malattie molto comuni quali l'artrite reumatoide e l'artrosi ; oppure a fratture e/o limitata capacità neuromotoria in seguito ad un'infarto.
In quest'ultimo caso, sono stati sviluppati specici trainings riabilitativi, per un recupero quasi totale delle funzionalità della mano.
La letteratura medica ed ingegnieristica contiene parecchi esempi di esercizi riabilitativi. Ad esempio, nel [12], è descritto come ai pazienti, dotati di un "wired-glove" montato sull'arto danneggiato, è stato richiesto di svolgere i tre seguenti trainings:

- Transverse volar grip: il paziente deve prendere una bottiglia da un tavolo, versarne il contenuto in un bicchiere ed inne riporre la bottiglia nella posizione iniziale sul tavolo;
- The spherical volar grip: il soggetto deve svitare un barattolo e riporre il coperchio sul tavolo;
- The pulp pinch: il paziente deve rimuovere un piolo da un pannello di forma quadrata ed inserirlo in uno dei buchi di un secondo pannello posto accanto al primo;

Per ciascun task, il punteggio del paziente è stato stabilito calcolando la correlazione tra la curva realizzata con i dati ottenuti dalla misurazione dei suoi movimenti, e quella realizzata sottoponendo agli stessi tests un soggetto sano, privo di qualunque disabilità. Di conseguenza, la caratterizzazione quantitativa dei movimenti della mano è di notevole importanza per quanticare il grado di disabilità delle mani. Pertanto, uno strumento obiettivo e preciso in grado di misurare e stimare la cinematica della mano, è necessario per la valutazione clinica.

2.1.2 Trainings riabilitativi e soluzioni sensorizzate

Le speciche cliniche del dispositivo "guanto", ossia gli esercizi che dovrà essere in grado di monitorare, sono stati determinati da un gruppo di ricercatori del dipartimento di Psicologia della Neuroscienza dell'Università di Torino (Italia). Tale gruppo è noto come SAMBA(Spatia Motor and Bodily Awareness); lo scopo principale del loro lavoro consiste nella riabilitazione di pazienti che hanno subito ferite e menomazioni alle mani. Questa riabilitazione è basata su esercizi giornalieri degli arti superiori, per l'allenamento contemporaneo di velocità, precisione e forza. Di seguito, sono stati descritti tali esercizi:

- Pinching: consiste nel mantenere fermo il pollice e toccarlo con le altre quattro dita, seguendo speciche sequenze che devono essere ripetute diverse volte;
- Grasping: il paziente deve aerrare un oggetto e rilasciarlo per un dato numero di tentativi;
- Wrist training : consiste nell'eseguire rotazioni e/o essioni/estensioni del polso, per allenare ed irrobustire le articolazioni intercarpali;
- Reaching: il paziente deve prendere un oggetto con la mano, nel minor tempo possibile;

® Following: dopo un segnale acustico, che determina l'inizio dell'esercizio, il paziente deve seguire con un dito (indicato dal dottore/sioterapista) un punto luminoso sullo schermo, nel modo più preciso possibile.

L'obiettivo del "sistema-guanto", che dev'essere progettato, consiste nel misurare tutte le quantità siche relative agli esercizi riabilitativi appena descritti. Per la misurazione di ciascuna grandezza sica sono usate tecniche e sensori dierenti:

® Pinching. I tempi di esecuzione e gli errori connessi durante lo svolgimento di speciche sequenze, possono essere misurati usando sensori di forza.
Per tempo di esecuzione s'intende l'intervallo tra l'emissione del segnale acustico (via software) e il contatto fra il pollice ed una fra le altre dita (a seconda della sequenza eseguita). Gli errori nella sequenza svolta, vengono determinati mettendo a confronto le sequenze corrette memorizzate nel pc, con i dati misurati all'uscita dei sensori di forza.

® Grasping.Anche in questo caso, sono sfruttati sensori di forza per la stima della forza esercitata dalle dita nell'aerrare un'oggetto. Gli angoli di apertura delle articolazioni PIP, invece, sono valutati facendo uso di sensori di bending.

® Wrist training. Per quanto riguarda, invece, i movimenti di rotazione e essione del polso, possono essere stimati per mezzo di "devices" come: accelerometro 3D, giroscopio 3D e magnetometro tri-assiale.
Il giroscopio permette di determinare le variazioni di orientazione spaziale della mano, integrando le velocità angolari misurate; mentre l'accelerometro permette la misurazione delle accelerazioni lineari, nota la massa "M" dell'oggetto accelerato (sono basati sulla seconda legge di Newton $F = M * a$). In pratica, è possibile risalire all'accelerazione "a", misurando la forza "F". Quest'ultima, può essere valutata misurando la deformazione di una molla alla quale è collegata la massa "M", attraverso un sensore strain-gauge o un sensore piezoelettrico,[83].
Il magnetometro è uno strumento usato per misurare la densità di usso magnetico B (unità di misura [Tesla] o $[\frac{A*s}{m^2}]$), prodotta dal campo magnetico Terrestre, [84].Poichè, la densità del usso magnetico nell'aria è direttamente proporzionale alla forza del campo magnetico H (unità di misura $\frac{A}{m}$]), un magnetometro è in grado di rilevare le uttuazioni nel campo della Terra.
Si possono distinguere, in generale, due tipi di magnetometro:

Magnetometri vettoriali : misurano il valore della densità di usso in una specica direzione dello spazio tridimensionale; la densità di usso, è denita pertanto come un vettore dotato di direzione, verso e intensità;

Magnetometri scalari : misurano soltanto l'intensità della densità di usso magnetico;

E' importante sottolineare, che nella maggior parte dei "wired-glove" l'informazione relativa alla posizione spaziale assoluta, è ottenuta come combinazione dei dati in uscita da giroscopi, accelerometri e magnetometri, in modo da ridurre gli errori causati dall'integrazione numerica e dal rumore sovrapposto alle misure stesse.

⑧ Reaching. La principale quantità misurata è il tempo di esecuzione dopo il segnale acustico.

⑧ Following. Le quantità da considerare sono due: il tempo di reazione del paziente nell'iniziare il movimento della mano; e il tempo di esecuzione , dunque tra l'inizio del movimento e il raggiungimento dello spot luminoso. Tali intervalli di tempo possono essere stimati per mezzo di un accelerometro, che rileva l'inizio e la ne del movimento stesso.
Tuttavia, oltre a tali grandezze meccaniche, il sioterapista/dottore dev'essere in grado di stabilire le condizioni siologiche del paziente. Infatti, durante lo svolgimento delle procedure riabilitative due importanti parametri tendono ad aumentare: la temperatura e il battito cardiaco.

Battito cardiaco e frequenza cardiaca. Il battito cardiaco indica i movimenti di contrazione ed espansione del cuore. La frequenza cardiaca è denita, appunto, come la misura dei battiti cardiaci al minuto (bpm = beat per minute). Queste due quantità sono solitamente stimate con un'unico sensore di pulsazione.

Temperatura. La temperatura è un'ottimo indicatore del movimento propiocettivo delle mani; si usa banalmente un sensore di temperatura per misurarla.

Un'altro parametro signicativo è la sudorazione. Durante lo svolgimento di un esercizio, il paziente potrebbe incominciare a sudare dalle mani. Questo è un'altro buon indicatore del corretto funzionamento del sistema propiocettivo del paziente. Una semplice soluzione, per il monitoraggio di tale grandezza, consiste nell'uso di un sensore GSR (Galvanic Skin Response).

Training	*Quantity to measure*	*Sensor*
PINCHING	Execution time	Force sensor
	Sequence errors	
GRASPING	Force of the fingers	Force sensors
	Opening angles of the fingers	Bend sensor
WRIST	Wrist rotation	3 axis Accelerometer, 3-axis Gyroscope, Magnetometer
	Wrist flexion	
REACHING	Execution time	Force sensor
FOLLOWING A SPOT ON A SCREEN	Reaction time	Accelerometer
	Execution time	Accelerometer
PHYSIOLOGICAL MEASUREMENTS	Heart Pulsation and Heart Rate	Pulsation sensor
	Temperature	Temperature sensor
	Sweating	Skin conductance sensor

Figura 2.2: Tabella trainings

2.2 Scheda prototipo

Il processo di design del "sistema-guanto" di tipo wireless, è iniziato dall'analisi della PCB-prototipo realizzata come tesi di laurea dallo studente L.Mastrototaro. Tale PCB è stata progettata come dispositivo di testing dei diversi sensori e componenti, che verranno poi montati sul "wired-glove nale".

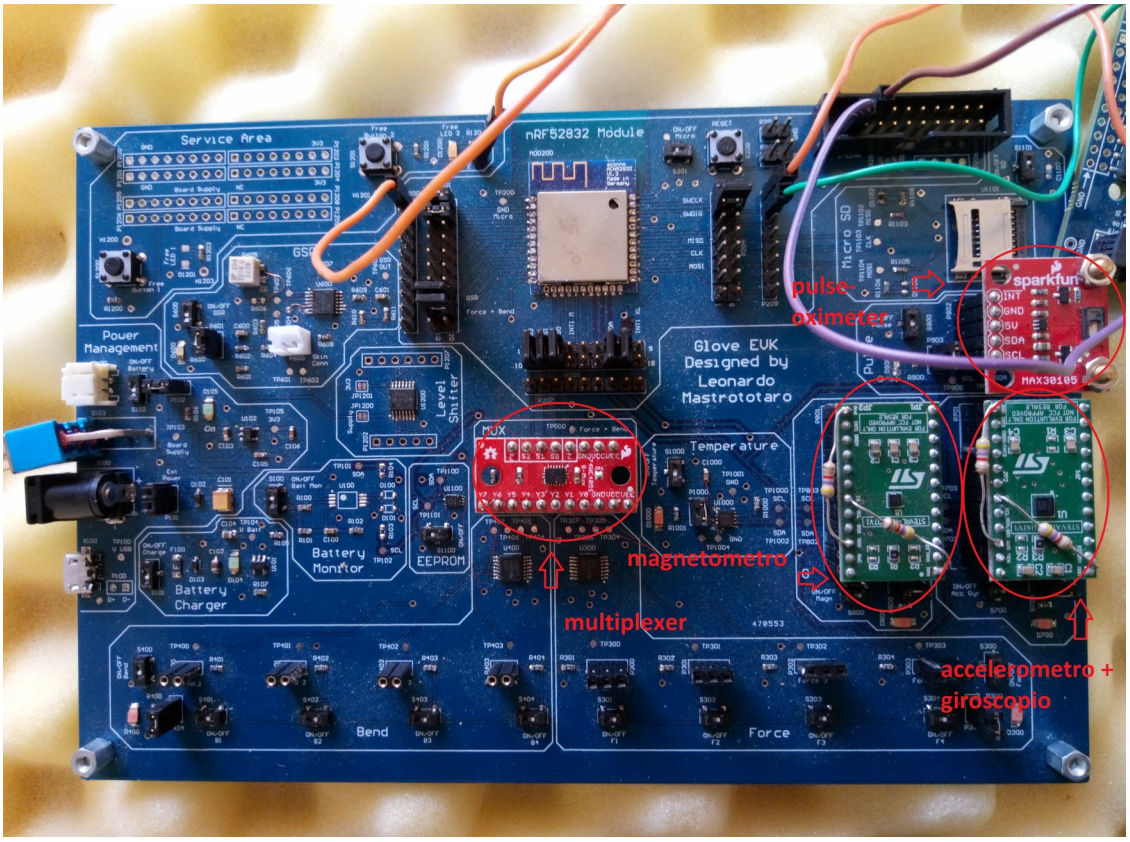


Figura 2.3: PCB-prototipo

La gura 2.3, mostra una scheda di dimensioni 190mm x 127mm e con spessore di circa 1mm, su cui sono state montate altre quattro mini-PCB (due rosse e due verdi). Quella rossa al centro integra un multiplexer analogico-digitale a 8 ingressi (dunque tre segnali di controllo), mentre le altre tre (sulla destra) integrano dei cosiddetti Application circuits relativi a : un sensore di tipo pulse-oximeter, un accelerometro-giroscopio tri-assiale ed un magnetometro (spesso sono chiamate Evaluation boards).Al di sopra del multiplexer è stato posizionato il microcontrollore Aconno basato sull'nRF52832, dotato di antenna bluetooth per la trasmissione dati tra scheda e pcNella parte inferiore della board sono stati collocati opportuni alloggiamenti per poter collegare rispettivamente, quattro sensori di bending e quattro sensori di forzaIl guanto nale sarà dotato, infatti, di un sensore di ciascun tipo per l'indice, il medio, l'anulare e il mignolo; mentre non sono previsti sensori di questo tipo per il pollice.
Sono stati inseriti, inoltre, un sensore di temperatura e un sensore GSR (Galvanic Skin Response).Quest'ultimo è usato per rilevare i cambiamenti delle proprietà elettriche della pelle, [76], in seguito ad eventi come lo stress o la sudorazione da esercizio sico. Il GSR è caratterizzato da una coppia di elettrodi posizionati su una coppia di dita (in genere l'indice e il medio), che permettono tramite una

misura indiretta la valutazione della resistenza o della conduttanza della pelle umana.
La scheda è completata da una memoria di tipo EEPROM per lo storage dei dati, e da una sezione di "Power Management".La PCB può essere alimentata sia collegandola al pc attraverso una micro-usb di tipo B, sia impiegando una batteria esterna con l'apposito connettore.

2.2.1 Descrizione dettagliata

E' stata eseguita un'analisi dettagliata della "scheda-prototipo", con lo scopo di comprendere il funzionamento di tutte le sue parti, in modo da mantenere gli elementi fondamentali per lo sviluppo del wired-glove nale; eliminando, invece, tutti componenti ridondanti e/o superui. Infatti, l'obiettivo nale di questo lavoro di tesi è stato di ridurre le dimensioni di questa PCB a valori tali da poterla collocare sul dorso di una mano.
La Glove EVK progettata da L.Mastrototaro può essere scomposta in cinque sezioni principali:

- Service Area;
- Power Management;
- Memorie;
- Microcontrollore;
- Sensori analogici e digitali ;

Power Management

L'area di Power Management è stata realizzata come un'apposito circuito per l'alimentazione dell'intera board. Questa EVK è stata pensata come un device di testing il più adattabile possibile, infatti la si può collegare alla sorgente di alimentazione in ben quattro modi possibili:

- 1)Attraverso una micro-usb tipo b; si può alimentare la scheda persino con un laptop.
- 2) Usando un connettore jack DC. Si tratta di un comune jack di tipo barrel per la connessione alle prese da muro. E' dotato di un foro di diametro 5.5 mm e di un polo centrale di diametro 2.1 mm.

® 3) Un connettore per le batterie ai polimeri di Litio (LiPo). Le LiPo sono batterie ricaricabili agli ioni di litio, ma che utilizzano un materiale elettrolitico polimerico invece del comune liquido elettrolita.

® 4) Un header maschio a 2 pins per la connessione di una generica tensione d'alimentazione esterna.

Figura 2.4: connettori, partendo da sinistra: 1) micro-usb; 2) jack barrel DC; 3) connettore LiPo battery; 4) header maschio 2 pins [77]

E' evidente che, a seconda del metodo di alimentazione scelto, si avranno valori di tensione diversi all'ingresso della board.
Analizzando i datasheets dei diversi componenti e sensori è stato scelto di uniformare la tensione distribuita sull'intera board ad un valore di 3.3 V. Di conseguenza, è stato inserito sulla scheda un dispositivo in grado di ricevere in ingresso un dato valore di tensione e di fornire in uscita esattamente 3.3 V : si tratta appunto di un regolatore di tensione.
E' stato scelto un regolatore di tensione lineare di tipo LDO (low-dropout), poichè permette di ottenere una regolazione molto precisa, anche quando la tensione al suo ingresso è molto prossima al valore che deve fornire in uscita. L'LDO selezionato è il REG113NA-3.3/3K della Texas Instruments.

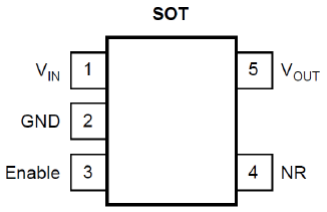


Figura 2.5: Regolatore LDO

La REG113 è una "famiglia" di regolatori lineari LDO con basso rumore in uscita e corrente ridotta sul pin di GND. La nuova topologia DMOS fornisce miglioramenti signicativi rispetto ai devices precedenti : ridotta caduta di tensione sul regolatore pari a 250 mV ; corrente tipica sul pin di GND di soli 850μA con una I_{OUT} = 400mA, e scende no a 10 μA quando il dispositivo non è abilitato.

Basandosi sul datasheet del componente, il circuito "stampato" sulla board è il seguente:

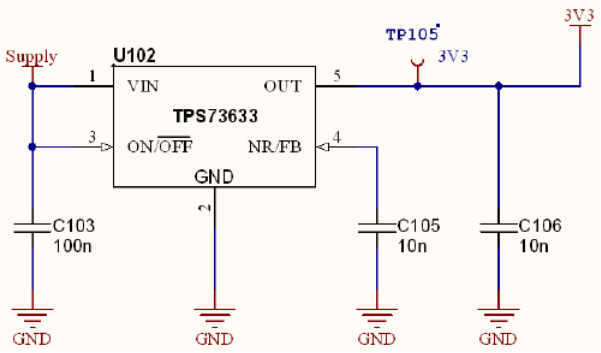


Fig 4.3. - Conditioning circuit of the LDO.

Figura 2.6: circuito LDO, [77]

Dalla gura 2.6, si nota come il pin Vin sia collegato alla sorgente di alimentazione esterna e il pin Out fornisce in uscita i 3.3 V (3V3).
Tenendo in considerazione l'opzione di collegare una batteria LiPo, occorre un circuito per poterla ricaricare. Per tale scopo, è stato usato l'MCP73831T-2ACI/OT prodotto dalla Microchip technology, mostrato nella gura seguente.

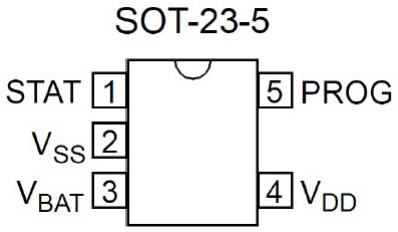


Figura 2.7: carica-batteria

Dopodichè è stato realizzato con Altium Designer, l'Application Circuit del dispositivo citato:

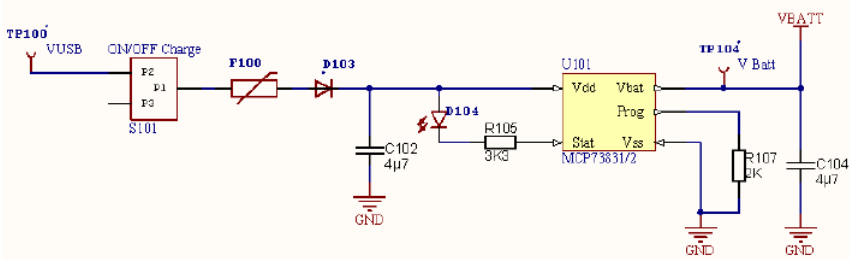


Figura 2.8: circuito del carica-batteria, [77]

L'ingresso è rappresentato dalla tensione proveniente da uno dei 3 connettori (tranne quello della batteria ovviamente) sopra elencatiDa notare che, è stato inserito un fusibile "F100" più un diodo "D103" come circuiti di protezione per eventuali sovratensioni che possono danneggiare l'MCP. Inoltre, è stato adottato un semplice switch "S101" per abilitare/disabilitare il funzionamento di questo circuito.
Dopodichè, è stato inserito un battery level indicator. In questo modo, infatti, è possibile sapere per quanto tempo la board può funzionare, quando viene alimentata tramite batteria. Il dispositivo scelto è stato il DS2782 prodotto dalla Maxim Integrated.

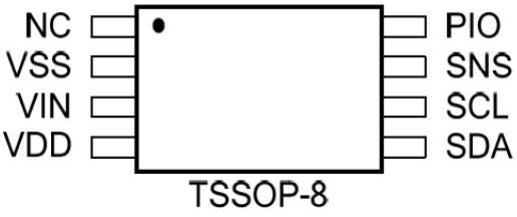


Figura 2.9: battery level indicator

E' un dispositivo basato sul protocollo I2C per la trasmissione dei dati, misurando tensione, temperatura e corrente; stimando, attraverso tali quantità, la capacità di carica ([mAh] o [Ah]) rimanente nelle batterie ricaricabili agli ioni di litio e di tipo LiPo.
Anche in questo caso è stato riportato il circuito realizzato, basandosi sull'Application circuit indicato sul datasheet del componente:

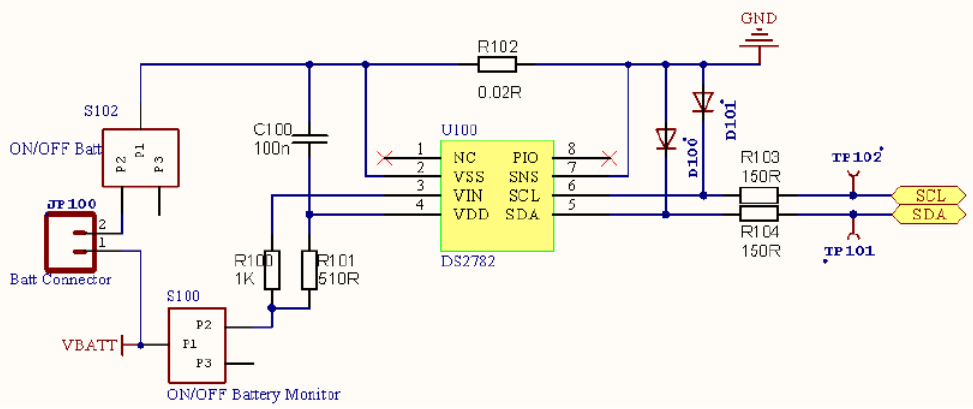


Figura 2.10: circuito indicatore livello batteria, [77]

In gura 2.10, si nota come sono stati aggiunti due switches:uno per abilitare l'alimentazione tramite batteria "S102", l'altro per attivare il circuito di monitoring del livello di carica della batteria stessa "S100".
Mettendo insieme i tre circuiti indicati: LDO regolatore di tensione, il battery charger e il battery monitor si ottiene la struttura nale della sezione di Power managemente:

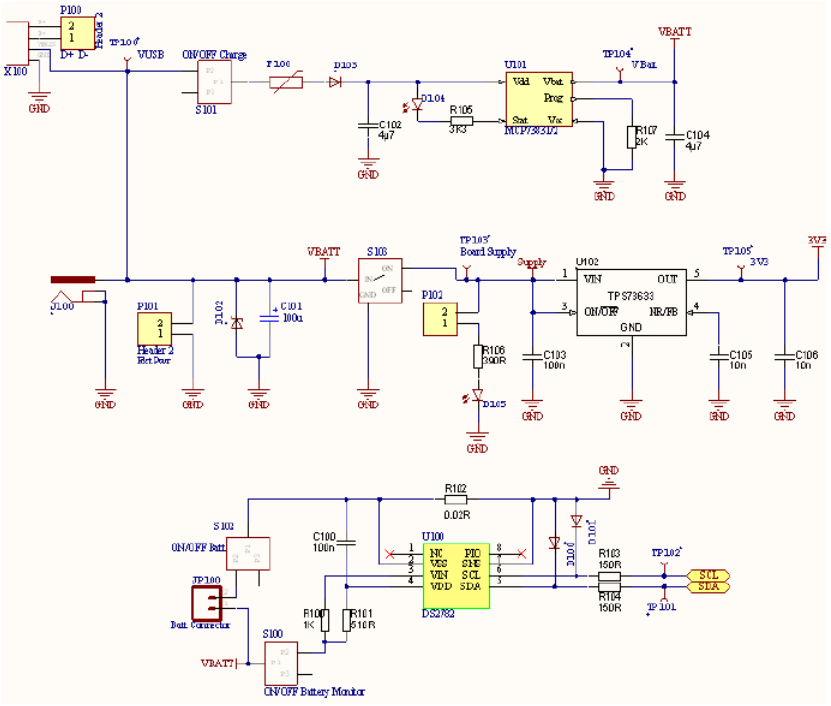


Figura 2.11: sezione di Power management, [77]

Al centro della gura 2.11, è possibile notare che è stato inserito un cosiddetto lever switch (S103) per poter disconnettere l'LDO da qualunque sorgente di alimentazione esterna e dunque permettere lo spegnimento della board stessa. Si nota, inoltre, l'inserimento di un LED (D105) per fornire l'informazione di board nello stato On all'utente nale. Inne, sono stati inseriti opportuni test points su quasi tutti i pins dei componenti citati, per favorire un rapido testing del loro funzionamento tramite multimetri e/o oscilloscopi.

2.2.2 Sensori analogici e digitali

Sensori di forza

Dopo un'attenta ricerca in letteratura, è stato deciso di scegliere un sensore di forza di tipo resistivo. Il suo principio di funzionamento si basa sulla riduzione

della resistenza all'aumentare della forza applicata.Questa tipologia di sensori è chiamata FSR (Force Sensing Resistor).
Gli FSR sono realizzati con un polimero conduttivo, che cambia il suo valore di resistenza, in modo prevedibile, a seconda della forza applicata sulla sua supercie. Normalmente, sono distribuiti nella forma di un foglio di polimero o come inchiostro applicato, su una supercie, con il processo di "screen-printing".Il lm di sensing, che riveste il sensore, consiste di sia particelle elettricamente conduttive sia non-conduttive sospese all'interno di una struttura a matrice.
Tali particelle sono di dimensioni al di sotto del micrometro e sono realizzate in modo da ridurre la dipendenza dalla temperatura incrementando le proprietà meccaniche e la resistenza superciale.
Applicando una forza alla supercie del lm di "sensing", provoca lo spostamento delle particelle, che toccano gli elettrodi conduttivi, cambiando la resistenza del lm.
Come la maggior parte dei sensori resistivi, i sensori di forza resistivi richiedono un'interfaccia piuttosto semplice e possono funzionare in modo soddisfacente in quasi qualunque tipo di ambiente.Se confrontati con altri tipi di sensori di forza, i vantaggi degli FSR sono :dimensioni ridotte (spessore tipicamente minore di 0.5 mm); costo ridotto e buona resistenza agli urti.Uno svantaggio, invece, è rappresentato dalla loro ridotta precisione:le misure prese possono variare fra loro anche sino al 10 %.Tuttavia, per la realizzazione di un primo dispositivo prototipale, risultano essere la scelta ottimale.
Sono stati considerati due tipi di FSR, piuttosto validi per la nostra applicazione: il Flexiforce A301 della Tekscan e l' FSR 402 della Interlink Electronics.Nella seguente tabella sono state messe a confronto le proprietà siche dei due sensori :

	FlexiForce A301	*FSR 402*
Thickness	0.20 mm (0.008 in.)	0.46 mm (0.0181 in.)
Length	25.4 mm (1 in.)	56 mm (2.2 in.)
Width	14 mm (0.55 in.)	14 mm (0.55 in.)
Sensing Area	9.5 mm (0.37 in.) diameter	12.7 mm (0.5 in.) diameter
Substrate	Polyester (ex: Mylar)	Polymer

Figura 2.12: proprietà siche sensori di forza, [77]

E' facile notare come le proprietà meccaniche siano molto simili fra loro.L'unica dierenza signicativa è nell' area di sensing : quella dell'A301 è minore di circa 3.2 mm rispetto a quella dell' FSR 402.
Per la nostra applicazione risulta più adatto un sensore con un'area di sensing più piccola. Infatti, un'area di sensing maggiore risulta si più adattabile ai diversi tipi di ngertips e dunque permette di misurare le dierenti pressioni esercitate dal

paziente. Tuttavia, le sue dimensioni potrebbero essere, più facilmente, causa di un errore di misura, rilevando pressioni non appartenenti al dito stesso. Inoltre, i produttori stessi raccomandano d'impiegare l'intera supercie di sensing per ottenere la miglior risposta del sensore. Pertanto, considerando il diametro medio delle dita della mano, il sensore che meglio si adatta al nostro wired-glove è l'A301.

	FlexiForce A301	*FSR 402*
Force Range	0 – 111 N	0 – 20 N
Resolution	Continuous (analogic)	Continuous (analogic)
Repeatability	± 2.5%	± 2%
Non-Actuated Resistance	> 10 MΩ	> 10 MΩ
Hysteresis	4.5 %	10 %
Response Time	< 5 µs	< 3 µs
Drift	< 5% per logarithmic time scale (constant load of 111N)	< 5% per logarithmic time scale (constant load of 98N)
Operating Temperature	-40°C – 60°C	-40°C – 85°C

Figura 2.13: Caratteristiche meccaniche dei sensori di forza [77]

Come mostra questa seconda tabella in gura 2.13, anche le caratteristiche meccaniche dei due sensori sono più o meno simili tranne che per il Force range e il valore d'Isteresi. Per Isteresi s'intende la deriva espressa in percentuale del valore in uscita dal sensore, applicando sempre la stessa forza in ingresso per un certo intervallo di tempo. E' importante, dunque, avere un basso livello d'isteresi. In secondo luogo, il range di forza della mano di un paziente, durante un processo riabilitativo, non supera i 25 N, pertanto il Flexiforce risulta essere più adatto. Come già descritto nel capitolo precedente, i sensori di forza sono usati durante l'esecuzione degli esercizi di grasping, pinching e di reaching. In particolare, risultano necessari quattro sensori di forza (per indice, medio, anulare e mignolo), poichè il pollice rimane sso durante i tre tipi di esercizi citati.

Sensori di Bending

La letteratura propone quattro tipi di sensori di bending: basati su lm piezoelettrico, sensori a bra ottica, sensori resistivi e sensori a eetto Hall. Nella nostra applicazione, i sensori di bending sono usati solo per il training di "grasping".
La quantità da misurare, infatti, sono gli angoli di apertura delle articolazioni PIP di indice, medio, anulare e mignolo. Per tale motivo, sono necessari solo quattro sensori collocati su queste dita. La disposizione dei sensori sul guanto, sarà molto simile a quella usata per i sensori di forza.
Escludendo i sensori a eetto Hall e i sensori basati su bra ottica, poichè piuttosto costosi, la nostra scelta dovrà essere fatta tra i sensori piezoelettrici e i

sensori resistivi. Tuttavia, la maggiorparte della letteratura studiata, predilige l'uso di sensori resistivi, soprattutto per la loro semplicità e i loro costi ridotti.
Questi tipi di sensori di bending sono passivi e tipicamente realizzati depositando uno strato d'inchiostro resistivo su un substrato plastico essibile, a forma di una sottile striscia di lunghezza compresa tra 1 inch (25.4 mm) e 5 inch (127 mm). A riposo, ossia quando la striscia è piatta (at condition), il sensore di bending è caratterizzato da una sua resistenza intrinseca. Non appena il sensore viene piegato, i materiali resistivi sono "stirati" e di conseguenza particelle adiacenti fra loro entrano in contatto, incrementando di conseguenza la resistenza. Tipicamente, la resistenza nominale (at condition) può variare tra 10kΩ e 50kΩ ed incrementa con l'aumentare della essione, no ad un massimo di un fattore 10.
Tra gli strati del substrato del sensore di bending, si trova uno "schema" stampato d'inchiostro conduttivo. Tale inchiostro contiene pigmenti di carbonio o argento che lo rendono, appunto, conduttivo. Di solito, le particelle di carbonio sono sospese all'interno dell'inchiostro per evitarne lo svanimento nel tempo.
La maggior parte dei sensori di bending basati su inchiostro conduttivo, sono di tipo "unipolare": ossia la loro resistenza cresce all'aumentare della essione del sensore in una specica direzione, mentre rimane invariata se se si applica una essione nella direzione opposta. Dunque, ponendo due dispositivi in congurazione "back to back", si riescono ad ottenere misure "bipolari", ossia in grado di misurare le essioni in entrambe le direzioni. Da notare come al crescere della lunghezza, la resistenza intriseca aumenta e di conseguenza anche la resistenza del sensore alla massima deessione.
Dopo un'attenta ricerca nella letteratura e tra i dispositivi presenti in commercio, è stato scelto il sensore di bending resistivo più diuso: il Bend Sensor prodotto dalla Flexpoint Sensor System.

Figura 2.14: Bend Sensor Flexpoint, [35]

Questo sensore è realizzato su un singolo strato, spesso all'incirca 5 mils (0.127 mm), di materiale essibile rivestito con un'inchiostro al polimero-carbonio. Tale

tipo di materiale resistivo è comunemente usato nella realizzazione diresistori (thick lm), potenziometri lineari e trasduttori. Da notare che il lm di substrato usato nel Flexpoint è di tipo plastico, spesso poliammide, rivestito poi dall'inchiostro conduttivo. Durante questo processo, piccole crepe vengono introdotte nel rivestimento polimero/carbonio. Tali crepe evitano la rottura del sensore, quando questo viene piegato anche ripetutamente per un certo intervallo di tempo. Inoltre, come detto il Bend sensor è realizzato su un singolo strato, dunque lo sporco, la polvere e altri tipi di particolato non ne inuenzano il corretto funzionamento, così come accadrebbe in strutture multistrato.
Nella gura seguente sono indicate le caratteristiche meccaniche del sensore di bending scelto:

Thickness	0.13 mm (0.005 in.)
Length	76.2 mm (3 in.)
Operating Temperature	-30 °C – 90 °C
Lamination	Polyester
Type	Unidirectional
Duration	> 10^6 times
Width	7 mm (0.28 in.)

Figura 2.15: Caratteristiche meccaniche, sensore bending, [77]

Di seguito, invece è indicata la caratteristica deessione-resistenza indicata sul datasheet:

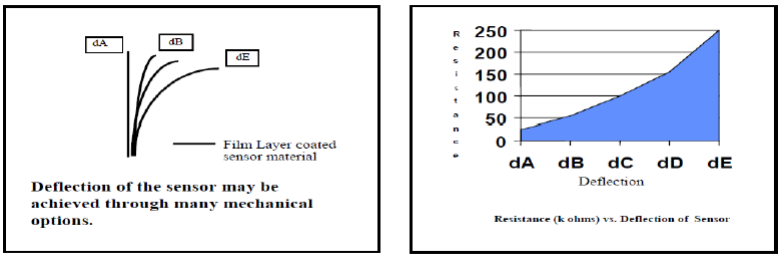


Figura 2.16: caratteristica deessione-resistenza, [35]

Il tasso di variazione della resistenza dipende principalmente da due fattori:

- il raggio di bending, più piccolo è il raggio maggiore sarà la variazione;
- la deessione angolare, ssato il raggio più il sensore è piegato maggiore sarà la variazione.

Sensore GSR

Il sensore GSR (Galvanic Skin Response) è usato per misurare e determinare di conseguenza, lo stato emotivo del paziente. Infatti, uno degli indicatori principali

che indicano l'aorare di un emozione, e che può essere quanticato è la risposta galvanica della pelle umana; nota anche come conduttanza della pelle (SC) o attività elettro-dermica (EDA), modula la quantità di sudore emessa dalle ghiandole sudoripare.Tale quantità è massima proprio in corrispondenza di mani e piedi.
Pertanto, la sudorazione, importante per una buona termo-regolazione corporea, provoca inoltre una variazione nella conduttanza della pelleTale variazione è spesso causata da una forte emozione provata, che causa un'incremento della conduttanza della pelle:da notare che sia un'emozione positiva, di gioia, sia un'emozione negativa, di tristezza, causa questo incremento. E' possibile aermare, dunque, che la conduttanza della pelle non subisce un controllo cosciente da parte dell'essere umano.
La conduttanza della pelle può essere misurata usando degli elettrodi posti sulle dita dei pazienti. Un sensore GSR, infatti, misura la conduttanza elettrica tra due punti, perciò può essere considerato come un particolare tipo di ohmetro.
La letteratura propone soluzioni diverse per misurare la resistenza della pelle tra due punti. Per esempio nel 2012 Tarabella, [85] insieme con altri ricercatori, ha sviluppato un sensore GSR basato sulla bra di cotone.Una singola bra di cotone è stata ricoperta con polietilenedioxytiofene dopato con polistirene sulfonato (PEDOT:PSS), quest'ultimo un polimero conduttivo, e usata come canale di un transistor organico elettromeccanico (OECT); tale bra s'interfaccia direttamente con un liquido elettrolitico, che realizza il dielettrico, il tutto è sormontato da un lo d'argento (Ag) che rappresenta il gate, si veda la gura seguente:

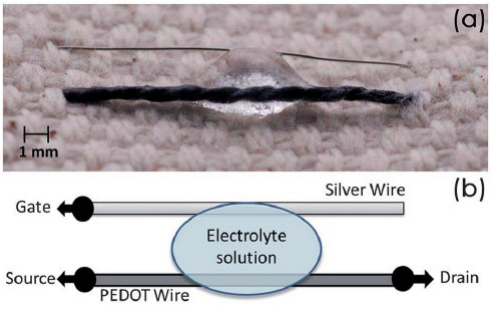


Figura 2.17: OECT transistor, [85]

Il dispositivo descritto è uno strumento ecace per la misurazione elettrochimica dell'NaCl nell'acqua. In generale, l'OECT risulta essere un dispositivo semplice e a basso costo, e di conseguenza molto usato in applicazioni di wearable electronics e di tipo medico.
Infatti, la determinazione della concentrazione salina del sudore emesso dal corpo umano è importante per stabilire il suo ruolo nella trasmissione degli impulsi

nervosi e nella contrazione muscolare. Inoltre, per quanto riguarda le applicazioni in ambito medico ("healthcare"), i sensori OECT al cotone potrebbero essere usati, ad esempio, per la diagnosi di malattie come la brosi cistica.
Un'altra soluzione è stata trovata da un gruppo di ricercatori di Bilbao (Spagna), che ha progettato un dispositivo GSR, [86], in grado di rilevare la dierenza di conduttanza della pelle quando una persona è sotto stress e quando non lo è, impiegando due elettrodi collocati su due dita; in questo modo il sensore si comporta come se ci fossero due terminali ai capi di una resistenza. Per determinare il valore di GSR, il dispositivo è dotato di una resistenza disposta in modo tale da formare un partitore di tensione insieme con la resistenza della pelle. Dopodichè, per ottenere una misura più accurata è inserito un ltro passa-basso, quindi l'uscita del ltro è inviata ad un microcontrollore, che garantisce la trasmissione dei dati via wireless tramite il protocollo ZigBee.

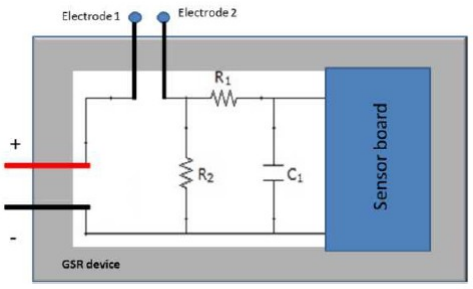


Figura 2.18: circuito condizionamento del GSR, [86]

Dalle misurazioni ottenute, è stato possibile osservare che la tensione d'uscita è inversamente proporzionale al valore della resistenza della pelle. Dunque, maggiore è lo stress provato dalla persona, maggiore la sudorazione delle sue mani, con conseguente riduzione della resistenza della pelle e aumento della tensione d'uscita misurata.
La ricerca portata avanti dal gruppo di Bilbao è quella che meglio si adatta al nostro wired-glove:dunque l'idea è quella di realizzare un dispositivo GSR che sfrutti due elettrodi da collocare su due dita della mano.Dopodichè, la tensione d'uscita misurata, dovrà essere ltrata con un opportuno circuito di condizionamento ed inviata al microcontrollore.
Anche in questo caso, dopo un'attenta ricerca tra i prodotti in commercio, è stato scelto il sensore proposto da SeedStudio Electronics della famiglia dei Grove sensors.Tale sensore funziona con una tensione di alimentazione sia di 3.3V sia di 5.0V, la sua sensitivity può essere regolata attraverso un potenziometro ed è dotato di due ditali da mettere sulle dita per eseguire le misurazioni.La gura seguente mostra il sensore GSR scelto:

Figura 2.19: Sensore GSR, SeedStudio, [87]

Sensore inerziale 3D (Accelerometro e Giroscopio)

L'uso di un'accelerometro e di un giroscopio triassiali nei trainings riabilitativi dei movimenti del polso e dell'inseguimento con il dito di uno spot luminoso sullo schermo.
L'esercizio riguardante la riabilitazione del polso è uno dei più dicili da portare a termine. Infatti, è necessario stimare due quantità:la rotazione del polso e la essione del polso. Tipicamente, le velocità angolari e le accelerazioni lineari massime di un'estremità del corpo umano, non superano rispettivamente i 500 [dps](degree per second) e gli 8 [g]. Quest'ultima considerazione è molto importante per la scelta del giusto dispositivo, tra la vastità di prodotti inerziali presenti sul mercato.
Alla ne è stato scelto l'LSM6DS0 prodotto dalla ST Microelectronics:

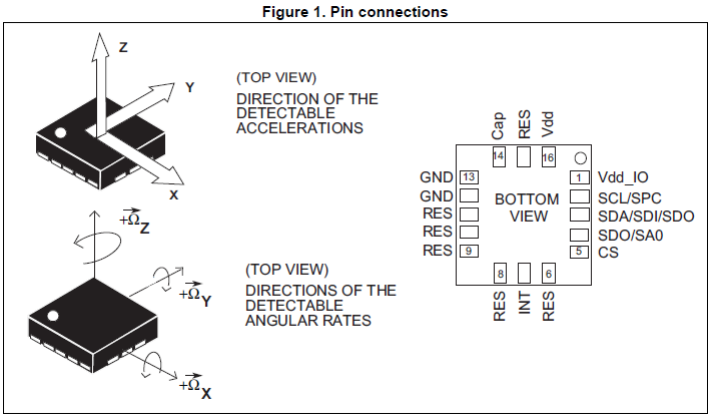


Figura 2.20: LSM6DS0 pinout, [79]

La famiglia dei sensori MEMS dell'ST, si ada sui robusti e adabili processi produttivi già usati nella realizzazione di giroscopi e accelerometri con la tecnica del micromachining. Mentre le interfacce integrate (IC) sono state sviluppate usando la tecnologia CMOS.

Come detto l'LSM6DS0 è un dispositivo "2-in-1", dal momento che racchiude sia un accelerometro 3D sia un giroscopio 3D. E' dotato, infatti, di tre canali d'uscita per le accelerazioni lineari e tre canali d'uscita per le accelerazioni angolari. Questo sensore ha due diversi modi di funzionamento: il primo in cui entrambi accelerometro e giroscopio sono attivi con lo stesso ODR(Output Data Rate), l'altro in cui è attivo soltanto l'accelerometro mentre il giroscopio si trova in "power-down" mode.
In questo SoC è stato integrato anche un sensore di temperatura. Per quanto riguarda la trasmissione dei dati da e verso il microcontrollore, è dotato di due interfacce seriali: I2C ed SPI.
Un parametro importante in questa tipologia di sensori è la Sensitivity sia dell'accelerometro lineare sia del giroscopio. Per quanto riguarda l'accelerometro lineare, la sua sensitivity può essere determinata, per esempio, applicando 1[g] di accelerazione al dispositivo. Ciò può essere fatto scegliendo come riferimento uno dei 3 assi cartesiani e collocando, di conseguenza, il chip su un piano ortogonale a tale asse, misurando il valore d'uscita in [LSB] dell'acceletometro; dopodichè si ruota il chip di 180deg e si misura nuovamente l'accelerazione. In questo modo è stata applicata un'accelerazione di +/- 1[g]. Dopodichè dividendo l'accelerazione d'ingresso applicata, per il risultato in uscita dal sensore digitale espresso in [LSB] si ottiene la sensitivity del sensore.
Il giroscopio, invece, è un sensore che produce un uscita digitale positiva per rotazioni in senso anti-orario intorno all'asse considerata. La sua sensitivity è denita come il guadagno del sensore.
Nelle seguenti tabelle, riportate dal datasheet del sensore, sono indicate, rispettivamente, le caratteristiche meccaniche ed elettriche dell'LSM6DS0:

2.1 Mechanical characteristics

@ Vdd = 2.2 V, T = 25 °C unless otherwise noted[3]

Table 3. Mechanical characteristics

Symbol	Parameter	Test conditions	Min.	Typ.[1]	Max.	Unit
LA_FS	Linear acceleration measurement range			±2		*g*
				±4		
				±8		
				±16		
G_FS	Angular rate measurement range			±245		dps
				±500		
				±2000		
LA_So	Linear acceleration sensitivity	FS = ±2 *g*		0.061		mg/LSb
		FS = ±4 *g*		0.122		
		FS = ±8 *g*		0.244		
		FS = ±16 *g*		0.732		
G_So	Angular rate sensitivity	FS = ±245 dps		8.75		mdps/LSb
		FS = ±500 dps		17.50		
		FS = ±2000 dps		70		
LA_TyOff	Linear acceleration typical zero-*g* level offset accuracy[2]	FS = ±8 *g*		±90		mg
G_TyOff	Angular rate typical zero-rate level[3]	FS = ±2000 dps		±30		dps
LA_ODR	Linear acceleration output data rate	Gyro ON		952 476 238 119 59.5 14.9		Hz
		Gyro OFF		952 476 238 119 50 10		Hz

Figura 2.21: Tabella1 : Caratteristiche meccaniche LSM6DS0, [79]

Table 3. Mechanical characteristics (continued)

Symbol	Parameter	Test conditions	Min.	Typ.[1]	Max.	Unit
G_ODR	Angular digital output data rate			952 476 238 119 59.5 14.9		Hz
Top	Operating temperature range		-40		+85	°C

1. Typical specifications are not guaranteed.
2. Typical zero-g level offset value after soldering.
3. Typical zero-rate level offset value after MSL3 preconditioning.

Figura 2.22: Tabella1 : Caratteristiche meccaniche LSM6DS0, [79]

2.2 Electrical characteristics

@ Vdd = 2.2 V, T = 25 °C unless otherwise noted

Table 4. Electrical characteristics

Symbol	Parameter	Test conditions	Min.	Typ.[1]	Max.	Unit
Vdd	Supply voltage		1.71		3.6	V
Vdd_IO	Power supply for I/O		1.71		Vdd + 0.1	V
LA_Idd	Accelerometer current consumption in normal mode	ODR = 10 Hz		60		µA
		ODR = 50 Hz		160		
		ODR ≥ 119 Hz		330		
G_Idd	Gyroscope current consumption in normal mode			4.0		mA
Top	Operating temperature range		-40		+85	°C
Trise	Time for power supply rising[2]		0.01		100	ms
Twait	Time delay between Vdd_IO and Vdd [2]		0		10	ms

1. Typical specifications are not guaranteed.
2. Please refer to *Section 2.2.1: Recommended power-up sequence* for more details.

Figura 2.23: Tabella 2: Caratteristiche elettriche LSM6DS0, [79]

Magnetometro

Durante l'esercizio di riabilitazione del polso risulta utile, inoltre, l'uso di un magnetometro. Infatti, un magnetometro triassiale fornisce l'angolo di rotazione spaziale rispetto ad uno dei quattro punti cardinali (O,E,N,S); angolo calcolato grazie alla misurazione delle tre componenti del campo magnetico terrestre ($\bar{B}_x, \bar{B}_y, \bar{B}_z$), che variano a seconda del luogo geograco in cui ci si trovaIn pratica, il magnetometro viene impiegato come bussola digitale.
Pertanto, combinando il magnetometro con l'accelerometro/giroscopio si è in grado di determinare la posizione spaziale della mano, durante gli esercizi svolti dal paziente. E' stato scelto il magnetometro LIS3MDL sempre della ST Microelectronics:

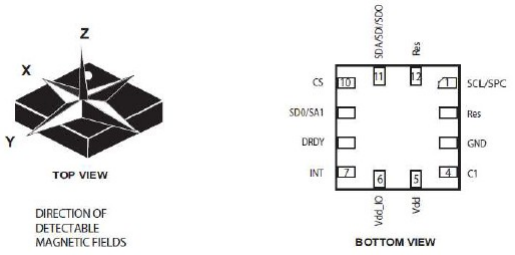


Figura 2.24: LIS3MDL pinout, [80]

Il LIS3MDL è un magnetometro a tre assi, ultra-low-power ad elevate prestazioni, che integra inoltre un sensore di temperaturaPuò essere congurato per svolgere un'operazione di self-test, che permette all'utente di vericare il corretto funzionamento del sensore all'interno dell'applicazione scelta.
Essendo un sensore digitale, anch'esso è dotato delle due interfacce seriali I2C ed SPI.
Particolarmente importanti da considerare, per questo tipo di sensore, sono gli errori di misura: infatti, una corrente che scorre in un lo conduttivo genera un campo magnetico (vedi legge di Biot-Savart); pertanto, nelle tracce di un circuito stampato si generano campi magnetici, quando scorre corrente, che si sommano al campo magnetico terrestre, causando errori nella valutazione del "compass heading". Dunque, è fondamentale collocare il sensore almeno qualche millimetro lontano dalle tracce in cui scorrono correnti superiori ai 10 mA.
Nelle seguenti gure sono state riportate le caratteristiche meccaniche ed elettriche del LIS3MDL:

@ Vdd = 2.5 V, T = 25 °C unless otherwise noted[a]

Table 3. Mechanical characteristics

Symbol	Parameter	Test conditions	Min.	Typ.[1]	Max.	Unit
FS	Measurement range			±4		gauss
				±8		
				±12		
				±16		
GN	Sensitivity	FS=±4 gauss		6842		LSB/ gauss
		FS=±8 gauss		3421		
		FS=±12 gauss		2281		
		FS=±16 gauss		1711		
Zgauss	Zero-gauss level	FS=±4 gauss		±1		gauss
RMS	RMS noise	X-axis; FS=±12 gauss; Ultra-high-performance mode		3.2		mgauss
		Y-axis; FS=±12 gauss Ultra-high-performance mode		3.2		mgauss
		Z-axis; FS=±12 gauss Ultra-high-performance mode		4.1		mgauss
NL	Non-linearity	Best-fit straight line FS = ±12 gauss Happlied = ±6 gauss		±0.12		%FS
ST	Self test[2]	X-axis FS = ±12 gauss	1		3	gauss
		Y-axis FS = ±12 gauss	1		3	
		Z-axis FS = ±12 gauss	0.1		1	
DF	Magnetic disturbance field	Zero-gauss offset starts to degrade			50	gauss
Top	Operating temperature range		-40		+85	°C

Figura 2.25: Tabella 3: Caratteristiche meccaniche LIS3MDL, [80]

@ Vdd = 2.5 V, T = 25 °C unless otherwise noted[c]

Table 5. Electrical characteristics

Symbol	Parameter	Test conditions	Min.	Typ.[1]	Max.	Unit
Vdd	Supply voltage		1.9		3.6	V
Vdd_IO	Power supply for I/O		1.71	1.8	Vdd+0.1	
Idd_HR	Current consumption in ultra-high-resolution mode	ODR = 20 Hz		270		µA
Idd_LP	Current consumption in low-power mode	ODR = 20 Hz		40		µA
Idd_PD	Current consumption in power down			1		µA
Top	Operating temperature range		-40		+85	°C

1. Typical specification are not guaranteed.

Figura 2.26: Tabella 4: Caratteristiche elettriche LIS3MDL,[80]

Pulsimetro

Il pulsimetro è usato per misurare la frequenza cardiaca del paziente, dopodichè con i dati raccolti è possibile plottare in tempo reale l'andamento del battito cardiaco. Un cuore funzionante è caratterizzato da pulsazioni o battiti regolari; l'obiettivo di questo tipo di sensore si concretizza nella misurazione dell'andamento dei battiti cardiaci, rilevando la velocità istantanea con cui il cuore si contrae e si espande, ossia i battiti per minuto(bpm) del paziente.
Solitamente, in ambito medico, il battito cardiaco è misurato monitorando le arterie in prossimità del polso o del collo.In questo caso, tuttavia, trattandosi di un sistema "wired-glove" è stato necessario denire un metodo di misura dierente. Le due soluzioni più ecenti trovate in letteratura sono le seguenti.
La prima è stata fornita dai ricercatori Rajala e Lekkala, [88], che hanno portato a termine uno studio comparato tra i sensori PVDF (realizzati con un lm di uoruro di polivinilidene) e i sensori basati su un lm elettromeccanico (EMFi).

I materiali appena citati garantiscono la generazione di una tensione d'uscita quando vengono deformati meccanicamente dall'applicazione di una forza esterna; in particolare Rajala e Lekkala hanno dimostrato come questi materiali possono essere usati nella misurazione di segnali siologici di tipo pulsato come la frequenza cardiaca e la respirazioneInoltre, sensori prodotti con materiali essibili e di spessore ridotto sono molto utili in applicazioni "wearable" di carattere medico, dove sono in contatto diretto con la pelle.Il sensore proposto, può essere integrato all'interno di vestiti o di oggetti di uso quotidiano, disponendo di una coppia di elettrodi miniaturizzati. In tal modo il sistema di misura risulterà essere poco ingombrante e piuttosto confortevole per i pazienti stessi.
Un'altro metodo adatto alla misurazione dei parametri cardiaci è basato sull'uso di un pulsimetro. Il pulsimetro è un sensore, di tipo non invasivo, adottato in ambito medico per la misurazione di frequenza cardiaca e ossigenazione sanguigna. La tecnica della pulso-ossimetria è stata, in origine, usata per monitorare la saturazione dell'ossigeno nei pazienti (SO_2). In particolare, poichè dalla gas analisi del sangue arterioso, spesso le due letture di SpO_2(saturazione periferica dell'ossigeno) e dell'SaO_2(saturazione arteriosa dell'ossigeno) non coincidono fra loro ed necessaria un'ulteriore misura di conferma, che è possibile eseguire con la pulso-ossimetria.Quindi, dai dati di saturazione dell'ossigeno è poi possibile ricavare informazioni riguardanti i parametri cardiaci.
Nelle sue applicaizioni più comuni, il pulsimetro è collocato su una parte sottile del corpo del paziente, solitamente un'estremità come la punta di un dito o il lobo di un'orecchio.
In quest'ottica, nel 2014 Lochner ed altri tre ricercatori hanno sviluppato un pulsimetro basato su materiali organici, che sono compatibili con substrati essibili:

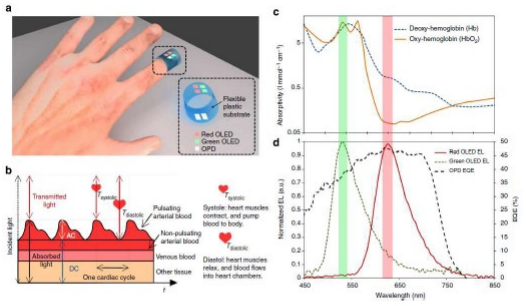


Fig 3.11. - Pulse oximetry with an organic optoelectronic sensor. (a) Pulse oximetry sensor composed of two OLED arrays and two OPDs. (b) A schematic illustration of a model for the pulse oximeter's light transmission path through pulsating arterial blood, non-pulsating arterial blood, venous blood and other tissues over several cardiac cycles. The a.c. and d.c. components of the blood and tissue are designated, as well as the peak and trough of transmitted light during diastole (Tdiastolic) and systole (Tsystolic), respectively. (c) Absorptivity of oxygenated (orange solid line) and deoxygenated (blue dashed line) haemoglobin in arterial blood as a function of wavelength. The wavelengths corresponding to the peak OLED electroluminescence (EL) spectra are highlighted to show that there is a difference in deoxy- and oxy-haemoglobin absorptivity at the wavelengths of interest. (d) OPD EQE (black dashed line) at short circuit, and EL spectra of red (red solid line) and green (green dashed line) OLEDs.

Figura 2.27: pulsimetro Organico, [89]

Il dispositivo consiste di due LEDs organici (OLED), uno verde (lunghezza d'onda d'emissione 532 nm) e uno rosso (lunghezza d'onda d'emissione 626 nm), più un fotodetector organico (OPD) sensibile alle lunghezze d'onda dei due LEDs. Gli strati attivi di cui è composto il sensore, sono stati depositati con la tecnica dello spin-coating.
La soluzione che meglio si adatta ai vincoli del nostro progetto è proprio quest'ultima, dunque la scelta di adottare un pulsimetro.Entrambe le soluzioni sono risultate non-invasive e poco costose, tuttavia il pulsimetro è più compatto e dunque più adatto ad essere collocato su parti del corpo come il polso o le dita.
Il pulsimetro scelto è il MAX30105 della MaximIntegrated :

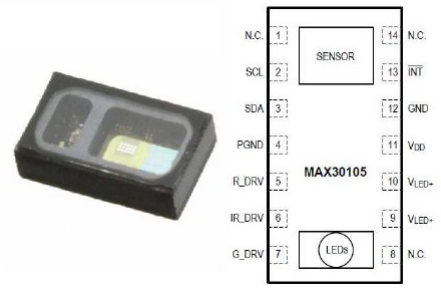


Figura 2.28: MAX30105 pinout, [90]

Il MAX30105 è un modulo integrato di "particle-sensing", che include al suo interno LEDs, fotodetectors ed un elettronica specica per la reiezione della luce dell'ambiente esterno.Grazie alle sue dimensioni ridotte, il MAX30105 può essere impiegato per applicazioni di tipo wearable; essendo un sensore digitale è dotato di un'interfaccia compatibile con lo standard I2C.
Il suo modulo ottico a 14 pins è racchiuso in un package di piccole dimensioni (5.6 mm x 3.3 mm x 1.55 mm) ; una copertura di vetro è usata per garantire la trasmissione e la ricezione dei segnali luminosi.
Il MAX30105 ha integrato, inoltre, un sensore di temperatura con risoluzione di 0.0625 deg (dunque no a quattro bits a destra del punto decimale).
Il sottosistema di "particle-sensing" è costituito dai seguenti elementisistema di cancellazione della luce dell'ambiente circostante (ALC), un convertitore analogico-digitale di tipo Sigma-Delta con 18 bits di risoluzione, un full-scale range programmabile tra $2\mu A$ e $16\mu A$, ed una frequenza di campionamento di $0.24MHz$.
Il MAX30105 è dotato di tre distinti LED drivers per pilotare rispettivamente un led verde (G), un led rosso (R) ed un led infrarosso (IR); la corrente di pilotaggio dei LEDs può essere programmata tra 0 e 50 mA, scegliendo l'opportuna tensione di alimentazione. Le gure seguenti riportano le caratteristiche elettriche del dispositivo:

Electrical Characteristics

(V_{DD} = 1.8V, V_{LED+} = 5.0V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at T_A = 25°C.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
POWER SUPPLY						
Power-Supply Voltage	V_{DD}	Guaranteed by RED and IR count tolerance	1.7	1.8	2.0	V
LED Supply Voltage	V_{LED+}	Guaranteed by PSRR of LED driver (R_LED+ and IR_LED+ only)	3.1	3.3	5.25	V
Supply Current	I_{DD}	Particle-sensing mode, PW = 215µs, 50sps		600	1100	µA
		IR only mode, PW = 215µS, 50sps		600	1100	
Supply Current in Shutdown	I_{SHDN}	T_A = +25°C, MODE = 0x80		0.7	2.5	µA
OPTICAL SENSOR CHARACTERISTICS						
ADC Resolution				18		bits
Red ADC Count (Note 3)	REDC	RED_PA = 0x0C, LED_PW = 0x01, SPO2_SR = 0x05, ADC_RGE = 0x00, T_A = +25°C		65536		Counts
IR ADC Count (Note 3)	IRC	IR_PA = 0x0C, LED_PW = 0x01, SPO2_SR = 0x05 ADC_RGE = 0x00, T_A = +25°C		65536		Counts
Green ADC Count (Note 3)	GRNC	GRN_PA = 0x24, LED_PW = 0x11, SPO2_SR = 0x05 ADC_RGE = 0x00, T_A = +25°C		65536		Counts
SNR IR LED	SNR_{IR}	White card loop-back, LED_PW = 0x11, ADC_RGE = 0x10, T_A = 25°C		89	300	dB
SNR Red LED	SNR_{RED}	White card loop-back, LED_PW = 0x11, ADC_RGE = 0x10, T_A = 25°C		88.9	300	dB
SNR Green LED	SNR_{GREEN}	White card loop-back, LED_PW = 0x11, ADC_RGE = 0x01, T_A = 25°C		80.4		dB

Figura 2.29: Tabella 5: caratteristiche elettriche MAX30105 parte 1

Electrical Characteristics (continued)

(V_{DD} = 1.8V, V_{LED+} = 5.0V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at T_A = 25°C.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Dark Current Count	LED_DCC	RED_PA = IR_PA = 0x00, LED_PW = 0x03, SPO2_SR = 0x01 ADC_RGE = 0x02			30	128	Counts
					0.01	0.05	% of FS
DC Ambient Light Rejection (Note 4)	ALR	ADC counts with finger on sensor under direct sunlight (100K lux), ADC_RGE = 0x3, LED_PW = 0x03, SPO2_SR = 0x01	Red LED		2		Counts
			IR LED		2		Counts
ADC Count—PSRR (V_{DD})	$PSRRV_{DD}$	1.7V < V_{DD} < 2.0V, LED_PW = 0x00, SPO2_SR = 0x05 T_A = +25°C			0.25	1	% of FS
		Frequency = DC to 100kHz, 100mV$_{p-p}$			10		LSB
ADC Count—PSRR (LED Driver Outputs)	$PSRR_{LED}$	3.6V < V_{LED+} < 5.0V, T_A = +25°C			0.05	1	% of FS
		Frequency = DC to 100kHz, 100mV$_{p-p}$			10		LSB
ADC Clock Frequency	CLK			10.2	10.48	10.8	MHz
ADC Integration Time (Note 4)	INT	LED_PW = 0x00			69		µs
		LED_PW = 0x01			118		
		LED_PW = 0x02			215		
		LED_PW = 0x03			411		
Slot Timing (Timing Between Sequential Channel Samples; e.g., Red Pulse Rising Edge To IR Pulse Rising Edge)	INT	LED_PW = 0x00			427		µs
		LED_PW = 0x01			525		
		LED_PW = 0x02			720		
		LED_PW = 0x03			1107		
COVER GLASS CHARACTERISTICS (Note 4)							
Hydrolytic Resistance Class		Per DIN ISO 719			HGB 1		
IR LED CHARACTERISTICS (Note 4)							
LED Peak Wavelength	λ_P	I_{LED} = 20mA, T_A = +25°C		870	880	900	nm
Full Width at Half Max	$\Delta\lambda$	I_{LED} = 20mA, T_A = +25°C			30		nm
Forward Voltage	V_F	I_{LED} = 20mA, T_A = +25°C			1.4		V
Radiant Power	P_O	I_{LED} = 20mA, T_A = +25°C			6.5		mW
RED LED CHARACTERISTICS (Note 4)							
LED Peak Wavelength	λ_P	I_{LED} = 20mA, T_A = +25°C		650	660	670	nm
Full Width at Half Max	$\Delta\lambda$	I_{LED} = 20mA, T_A = +25°C			20		nm
Forward Voltage	V_F	I_{LED} = 20mA, T_A = +25°C			2.1		V

Figura 2.30: Tabella 5: caratteristiche elettriche MAX30105 parte 2

Radiant Power	P_O	I_{LED} = 20mA, T_A = +25°C		9.8		mW
PARAMETER	**SYMBOL**	**CONDITIONS**	**MIN**	**TYP**	**MAX**	**UNITS**
GREEN LED CHARACTERISTICS (Note 4)						
LED Peak Wavelength	λ_P	I_{LED} = 50mA, T_A = +25°C	530	537	545	nm
Full Width at Half Max	$\Delta\lambda$	I_{LED} = 50mA, T_A = +25°C		35		nm
Forward Voltage	V_F	I_{LED} = 50mA, T_A = +25°C		33		V
Radiant Power	P_O	I_{LED} = 50mA, T_A = +25°C		17.2		mW
PHOTODETECTOR CHARACTERISTICS (Note 4)						
Spectral Range of Sensitivity	λ > 30% QE	QE: Quantum Efficiency	640		980	nm
Radiant Sensitive Area	A			1.36		mm^2
Dimensions of Radiant Sensitive Area	L x W			1.38 x 0.98		mm x mm
INTERNAL DIE TEMPERATURE SENSOR						
Temperature ADC Acquisition Time	T_T	T_A = +25°C		29		ms
Temperature Sensor Accuracy	T_A	T_A = +25°C		±1		°C
Temperature Sensor Minimum Range	T_{MIN}			-40		°C
Temperature Sensor Maximum Range	T_{MAX}			85		°C
DIGITAL INPUTS (SCL, SDA)						
Input Logic-Low Voltage	V_{IL}				0.3 x V_{DD}	V
Input Logic-High Voltage	V_{IH}		0.7 x V_{DD}			V
Input Hysteresis	V_{HYS}			0.5 x V_{DD}		V
Input Leakage Current	I_{IN}			±0.1	±1	µA
Input Capacitance	C_{IN}			10		pF
DIGITAL OUTPUTS (SDA, $\overline{INT}$)						
Output Low Voltage	V_{OL}	I_{SINK} = 3mA			0.4	V
I²C TIMING CHARACTERISTICS						
I²C Write Address				AE		Hex
I²C Read Address				AF		Hex
SCL Clock Frequency	f_{SCL}	Lower limit not tested	0		400	kHz
Bus Free Time Between STOP and START Condition	t_{BUF}		1.3			µs

Figura 2.31: Tabella 5: caratteristiche elettriche MAX30105 parte 3

(V_{DD} = 1.8V, V_{LED+} = 5.0V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at T_A = 25°C.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Hold Time (Repeated) START Condition	$t_{HD,STA}$		0.6			µs
SCL Pulse-Width Low	t_{LOW}		1.3			µs
SCL Pulse-Width High	t_{HIGH}		0.6			µs
Setup Time for a Repeated START Condition	$t_{SU,STA}$		0.6			µs
Data Hold Time	$t_{HD;DAT}$		0		0.9	µs
Data Setup Time	$t_{SU;DAT}$		100			ns
Setup Time for STOP Condition	$t_{SU;STO}$		0.6			µs
Pulse Width of Suppressed Spike	t_{SP}				50	ns
Bus Capacitance	C_b				400	pF
SDA and SCL Receiving Rise Time	T_r	(Note 5)	20		300	ns
SDA and SCL Receiving Fall Time	t_{Rf}	(Note 5)	20 x V_{DD}/5.5		300	ns
SDA Transmitting Fall Time	t_{of}		20 x V_{DD}/5.5		250	ns

Note 2: All devices are 100% production tested at T_A = +25°C. Specifications over temperature limits are guaranteed by Maxim Integrated's bench or proprietary automated test equipment (ATE) characterization.

Figura 2.32: Tabella 5: caratteristiche elettriche MAX30105 parte 4

Sensore di temperatura

Inne, un sensore di temperatura è stato introdotto nel nostro sistema "wired-glove", poichè la temperatura rappresenta un'ottimo indicatore dei meccanismi propriocettivi all'interno della mano.
Per la nostra applicazione è importante rispettare la specica sul range di temperatura del sensore, che dev'essere compatibile con le temperature medie del corpo umano: dunque tra i 25deg *C* e i 35deg *C*. Tuttavia, il mercato è pieno di dispositivi che rispettano tale intervallo, occorre pertanto restringere il campo delle possibili soluzioni introducendo un'ulteriore vincolo:elevata sensitivity.

Alla ne è stato scelto il sensore MAX30205 della MaximIntegrated :

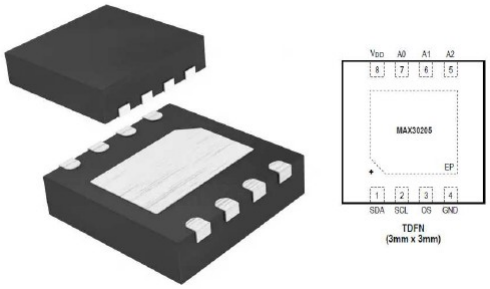


Figura 2.33: MAX30205 pinout, [81]

Questo sensore funziona correttamente nel range di temperatura tra 0 e 50 °C, e i valori digitali in uscita dall'ADC sono su 16 bits in complemento a 2, di cui 8 bits per la parte intera e 8 bits per la parte frazionaria dopo il punto decimale.L'ADC usato è un "Sigma-Delta". L'accuratezza del sensore, pari a 0.1°C soddisfa la specica termometrica clinica imposta dallo standard ASTM E1112 (Standard Specication for Electronic Thermometer for Intermittent Determination of Patient Temperature).
La comunicazione dei dati digitali in uscita, può avvenire sfruttando un'interfaccia compatibile con lo standard I2C. Inoltre, si tratta di un sensore perfettamente integrabile in un "wired-glove", grazie al package "TDFN" a 8 pins di dimensioni 3mm x 3mm.
E' importante specicare, che il MAX30205 misura la temperatura del suo die interno; dunque il cammino termico tra il die e l'esterno determina l'accuratezza della misura di temperatura. Poichè, questo sensore sarà saldato su una PCB, la temperatura misuratà sarà quella della PCB, che verrà rilevata dal pad scoperto sul fondo del sensore, per poi raggiungere il die interno. Perciò, il modo migliore per rilevare la temperatura ambiente, consiste nel montare questo sensore su una sezione della PCB che si trova a temperatura ambiente.
Nella seguente tabella sono mostrate le caratteristiche elettriche del MAX30205 :

Electrical Characteristics

(V_{DD} = 2.7V to 3.3V, T_A = 0°C to +50°C, unless otherwise noted. Typical values are V_{DD} = 3.0V, T_A = +25°C.) (Note 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Thermometer Error (Note 5)	T_{ERR}	0°C to +15°C	-0.5		+0.5	°C
		+15°C to +35.8°C	-0.3		+0.3	
		+35.8°C to +37°C	-0.2		+0.2	
		+37°C to +39°C	-0.1		+0.1	
		+39°C to +41°C	-0.2		+0.2	
		+41°C to +45°C	-0.3		+0.3	
		+45°C to +50°C	-0.5		+0.5	
ADC Repeatability	T_{repeat}	1 Sigma		0.009		°C
Temperature Data Resolution				16		Bits
Conversion Time			44		50	ms
First Conversion Completed		Data ready after POR			50	ms

Figura 2.34: Tabella 6: caratteristiche elettriche MAX30205 parte 1

Electrical Characteristics (continued)

(V_{DD} = 2.7V to 3.3V, T_A = 0°C to +50°C, unless otherwise noted. Typical values are V_{DD} = 3.0V, T_A = +25°C.) (Note 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Quiescent Supply Current	I_{DD}	I²C inactive, T_A = 0°C to +50°C		600	925	µA
		Shutdown mode, I²C inactive, T_A = 0°C to +50°C		1.65	3.5	
OS Delay		Depends on fault queue setting	1		6	Conversions
T_{OS} Default Temperature		Factory default setting		80		°C
T_{HYST} Default Temperature		Factory default setting		75		°C
POR Voltage Threshold				2.26		V
POR Hysteresis				130		mV
Input-High Leakage Current	I_{IH}	V_{IN} = 3.3V (all digital inputs)		0.005	1	µA
Input-Low Leakage Current	I_{IL}	V_{IN} = 0V (all digital inputs)		0.005	1	µA
Input Capacitance		All digital inputs		5		pF
Output-High Leakage Current		V_{IN} = 3.3V (SDA and OS)			1	µA
OS Output Saturation Voltage		I_{OUT} = 4.0mA			0.8	V
Output Low Voltage		I_{OL} = 3mA (SDA)			0.4	V

Figura 2.35: Tabella 6: caratteristiche elettriche MAX30205 parte 2

2.2.3 Service Area

La cosiddetta "area di servizio" racchiude un'insieme di componenti che introduce funzioni utili e aggiuntive alla scheda prototipo. Tuttavia, gran parte di questi elementi verranno eliminati per la realizzazione della scheda miniaturizzata, per rispettare il vincolo principale dell'area.
Sono stati introdotti in questa sezione della board:una coppia di headers femmina da 8 pins connessi a GND; una coppia di headers femmina da 8 pins connessi a 3.3V; una coppia di headers femmina da 8 pins connessi a "SUPPLY" (ossia la tensione di alimentazione fornita dall'esterno) e una coppia di headers femmina da 8 pins lasciati "unconnected" (NC). Questo insieme di headers collegati ai principali livelli di tensione della board svolgono un ruolo precisofornire una zona di "testing points" per la verica di eventuali errori riguardanti le connessioni verso le tensioni di alimentazione.
In questa zona di servizio, sono stati inoltre aggiunti due "buttons" tattili MJTP1230 della APEM Inc e due LEDs. Un LED (LED free 1) è semplicemente connesso a GND attraverso una resistenza serie; l'altro, invece, (LED free 2) è connesso ad un header maschio da 1 pin, in questo modo può essere pilotato dal microcontrollore durante la fase di testing della board prototipo.

Figura 2.36: Tactile button MJTP1230 della Apem Inc., [92]

Inne, è stato inserito un level shifter in questa sezione della board.Il level shifter è un dispositivo che converte i segnali digitali da uno standard logico ad un'altro

(esempio : nella connessione di porte logiche della famiglia TTL con quelle dellla famiglia CMOS). Spesso, è anche noto come "translator", poichè connette un circuito digitale che funziona su un dato livello logico ad un'altro che impiega un dierente livello logico. Da notare, che si tratta di un dispositivo diverso da un convertitore DC-DC o un regolatore di tensione, poichè la potenza consumata è inferiore e il suo scopo è la trasmissione dell'informazione e non della potenza. E' stato scelto, per questa board, il level shifter ADG3304 prodotto dalla Analog Devices. Si tratta di un dispositivo bidirezionale dotato di quattro canali distinti. I pins V_{CCA} e V_{CCY} sono stati collegati rispettivamente alla tensione "SUPPLY" e alla tensione "3V3".

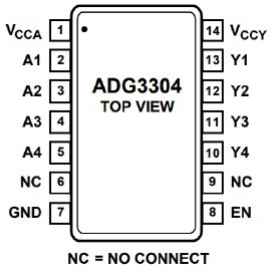


Figura 2.37: ADG3304 pinout , [91]

Lo schematico, realizzato con il software Altium Designer, della "Service Area" è riportato nella seguente gura:

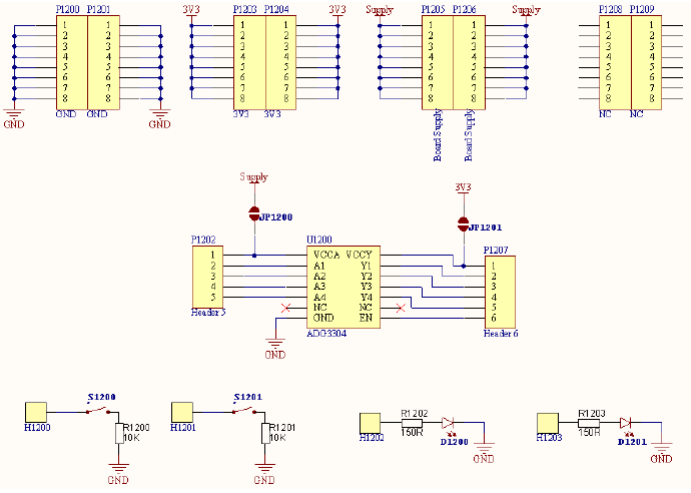


Figura 2.38: Service Area : schematic view, [77]

2.2.4 Microcontrollore

Una regione importante della scheda prototipo è quella occupata dal microcontrollore.

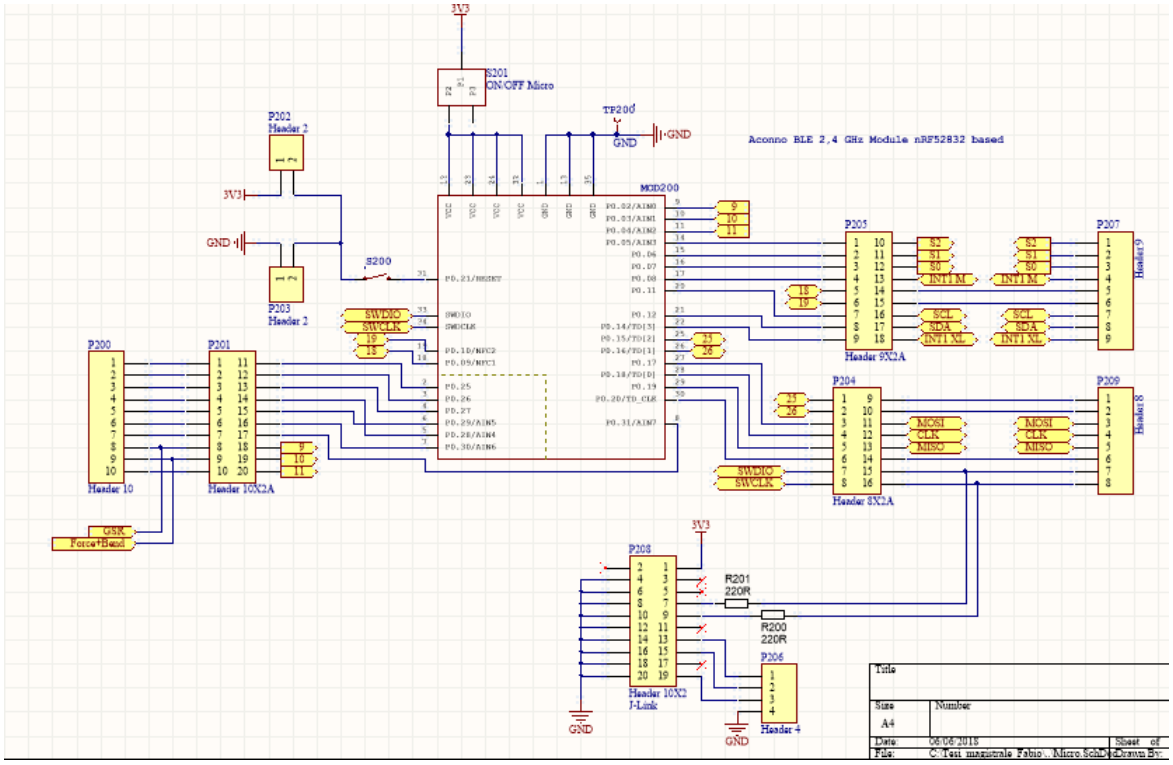


Figura 2.39: Schematic view microcontrollore, [77]

Il microcontrollore rappresenta il cervello della scheda, infatti tutti i segnali verso i componenti attivi provengono da quest'ultimo.
La tensione di alimentazione è fornita dalla linea 3V3 attraverso uno switch a 2 posizioni. Sul datasheet del microcontrollore nRF52832 è indicato il PO.21 come pin di RESET, che è stato connesso a GND attraverso un "tactile button" del tipo MJTP1230. Sui lati destro e sinistro del microcontrollore, è possibile osservare diversi segnali verso i sensori digitali e analogici. Come è facile notare dalla gura, 2.39, sono stati inseriti headers maschio sia a due righe sia a singola riga. Gli headers a due righe (vedi P201, P204 e P205) sono usati per connettere/disconnettere i pins del microcontrollore dal resto della board. La connessione/disconnessione è ottenuta tramite un female cap che è inserito sopra 2 pins adicenti dell'header maschio considerato.

Figura 2.40: Female cap, [77]

Invece, gli headers maschio a singola riga possono essere usati sia come "test points" a cui connettere un'oscilloscopio (attraverso cavi a coccodrillo) sia per connettere dei cavi "jumpers" e realizzare la programmazione del microcontrollore.

Figura 2.41: Crocodile clamp, [77]

Inoltre, il debugging della scheda prototipo è reso possibile anche grazie ad un header a due righe (P208) in cui verrà saldato il connettore maschio di tipo J-link :

Figura 2.42: Connettore J-link

Un connettore di tipo J-link, è un dispositivo che una volta connesso tra la board e il PC, permette la programmazione ed il debugging del microcontrollore impiegando uno specico software fornito dalla Nordic SemiconductorI pins del microcontrollore connessi al J-link sono SWDIO e SWDCLK.

LEDs

In corrispondenza di ciascun componente attivo è stato inserito un LED, che può essere abilitato/disabilitato tramite un'apposito switch. Il LED è usato per indicare lo stato del componente, se attivo o noI LEDs usati sono prodotti dalla Wurth Electronics:

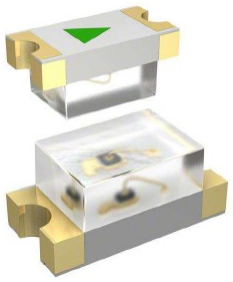


Figura 2.43: LEDs by Wurth Electronics

Per questa board prototipo, come circuito di pilotaggio dei LEDs è stata scelta una semplice resistenza serie tra la tensione di alimentazione (Vcc) e il nodo di anodo del LED. La resistenza serie è stata opportunamente scelta per far uire nel LED una corrente tale da non danneggiarlo:

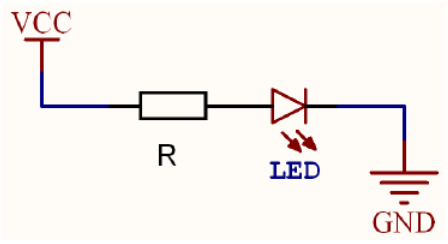


Figura 2.44: Circuito pilotaggio LED, [77]

Dopo aver analizzato il datasheet del LED, si è determinato il corretto valore della resistenza serie, considerando che si ha un corretto funzionamento di questo dispositivo per : una corrente pari a 10 mA e una tensione ai suoi capi pari a 2V. Di conseguenza è facile calcolare il valore del resistore da inserire in serie:

$$R = \frac{V_{cc} - V_{LED}}{I_{LED}} = \frac{V_{cc} - 2V}{10mA}$$

Poichè Vcc è stata scelta pari a 3.3V, R risulterà pari a 130Ω; il resistore con valore più prossimo è 150Ω (Scegliendo un valore più grande si ha una corrente di circa 8.67mA e dunque <10mA garantendo dunque il corretto funzionamento del

LED). Il valore di resistenza R è stato scelto sia per i LEDs dei componenti attivi (led rossi), sia per i LEDs della "Service Area" (led gialli).
Invece, per i LEDs verdi collocati nella sezione di Power Management e nella zona del Battery Charger, sono alimentati con Vcc pari a 5.5V, pertanto il valore calcolato per R risulterà pari a 350Ω. Tra i resistori in commercio è stato considerato quello di valore pari a 390Ω, sempre per il motivo di ottenere un valore di corrente ridotto.
Ciascun LED è collegato ad uno switch attraverso un header maschio a 2 pins connesso in serie, in modo tale da poter disabilitare il passaggio della corrente, nel caso venga realizzato un test sul consumo di corrente del componente attivo.

Capitolo 3

Capitolo 3: Progettazione PCB miniaturizzata

3.1 Design PCB miniaturizzata

Basandosi sul prototipo appena descritto, l'obiettivo di questo lavoro di tesi è stato realizzare una PCB miniaturizzata in modo da poter essere collocata sul dorso di una mano di dimensioni medie.Infatti, la tipologia di wired-glove, che vuole essere realizzata con questo progetto è innovativa.
Dopo un'attento studio e una conseguente analisi della letteratura descritta nel Capitolo 1, sono stati individuati due parametri principali per il design di questo guanto: primo, integrare le interconnessioni di rame tra la PCB e i sensori di bending e di forza, che saranno collocati sulle dita del guanto, usando pertanto materiali di tipo e-textiles per la fabbricazione del guanto stesso; secondo consiste nella progettazione di una PCB di dimensioni ridotte su cui integrare i sensori e il microcontrollore usati sulla scheda-prototipo, eliminando le parti superue come la "Service Area" e parte della sezione di "Power Management", 2.3.

3.2 Progettazione della scheda

Il design della scheda miniaturizzata è stato realizzato impiegando il software Altium Designer, specico per la progettazione di PCB. Iniziando, come descritto nel Capitolo precedente, dall'analisi della scheda prototipo progettata da Leonardo Mastrototaro; il primo step di design è stato basato sull'eliminazione delle sezioni superue della board, per rispettare il vincolo principale legato all'area.

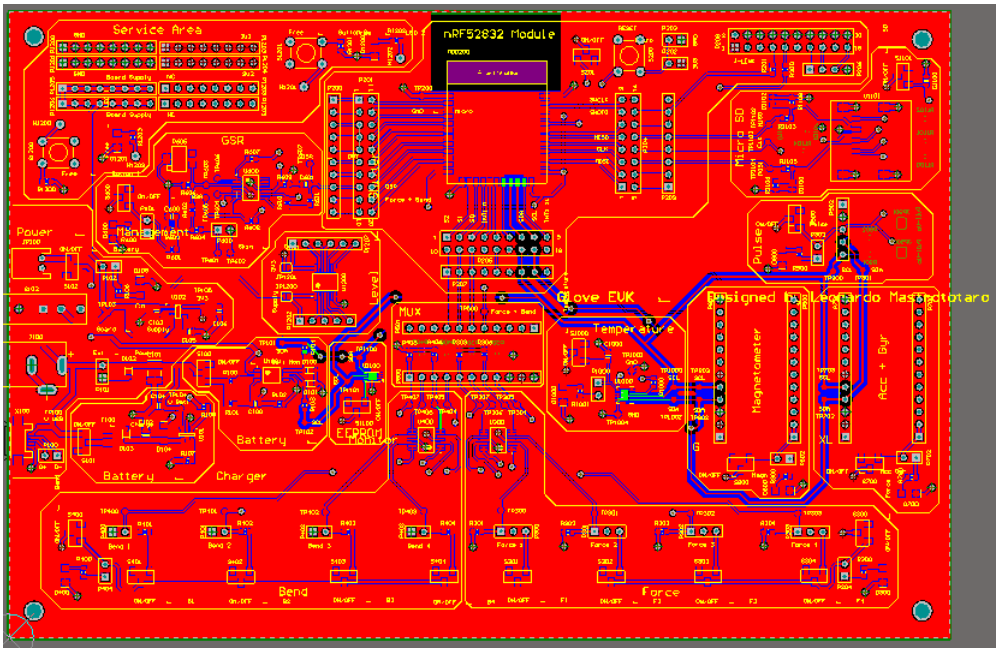


Figura 3.1: Evaluation board, Top Layer, designed by Leonardo Mastrototaro

La prima sezione eliminata è stata la Service Area, dunque sono stati eliminati i due free LEDs gialli, i due push-button MJTP1230 e tutti gli otto headers femmina usati per il testing. Si tratta di elementi aggiuntivi e non necessari che occupano uno spazio troppo grande; gli stessi buttom tattili risultano inutili, verrà inserito infatti un solo buttom tattile per il Reset della board, ma sarà scelto di dimensioni inferiori come descritto in seguito.
Dopodichè sono state studiate le sezioni riguardanti l'alimentazione della board (Power Management), la ricarica e il monitoraggio dello stato di carica della batteria (rispettivamente Battery charger e Battery monitor).
Iniziando dalla sezione di Power Management, come è stato descritto nel Capitolo 2, questa è dotata di ben quattro diversi tipi di connettori per ricevere la tensione d'alimentazione o da una wallplug (presa a muro), o da un pc, o da una batteria esterna di tipo LiPo o da un alimentatore DC-DC. Per la versione miniaturizzata della board sono stati mantenuti soltanto due connettori:il micro-usb tipo b e il connettore per la batteria LiPo.
Nella scheda prototipo è stato scelto di alimentare i componenti digitali e il microcontrollore stesoo con una tensione di 3.3V. Pertanto, è stato inserito un regolarore di tensione lineare di tipo LDO, il REG113NA-3.3/3K della Texas Instruments,2.6, ottenendo una tensione d'uscita pressochè costante e pari a 3.3V.
Dopodichè, considerando la board alimentata da una batteria LiPo esterna, è stato adottato un circuito per la ricarica della batteria, il MCP73831T-2ACI/OT della Microchip Technology, 2.7Inne, è stato scelto un circuito in grado di monitorare

lo stato di carica(SoC) della batteria, in modo da determinare la durata massima del funzionamento della board quando non alimentata con connessione wired. Battery monitor usato è il DS2782 della Maxim Integrated, 2.9.
Dopo un'oculata ricerca tra i numerosi dispositivi presenti in commercio, è stato scelto un "device" in grado di svolgere tutte e tre le funzioni degli integrati appena menzionati, in un unico chip. Si ha un notevole vantaggio in termini di area occupata, con l'unico svantaggio dell'aumento di complessità. Il chip scelto è il TPS65721 della Texas Instruments:

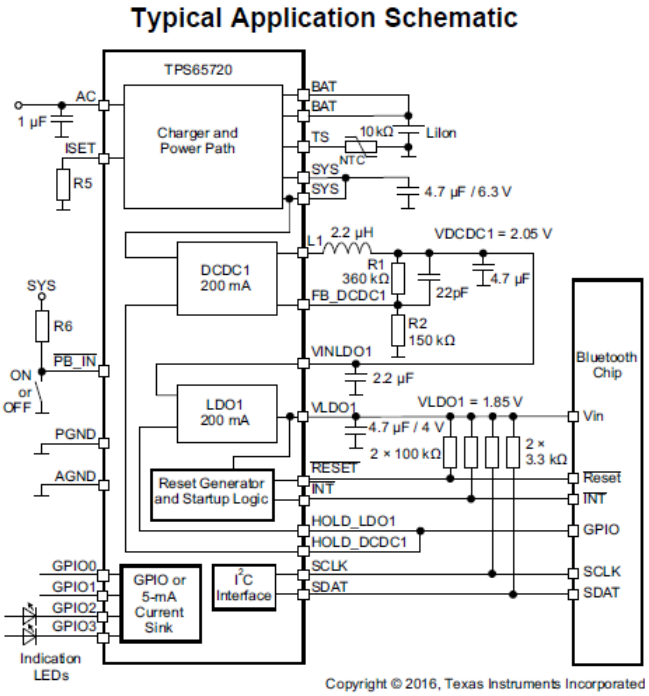


Figura 3.2: TPS65721 : Typical application circuit, [93]

In gura, 3.2, è facile distinguere i blocchi principali di cui è costituito questo dispositivo di power management: il Charger and Power Path, il DCDC1 converter (200 mA indica la massima corrente che può essere fornita in uscita), l'LDO1 converter e un blocchetto per pilotare quattro pins di GPIO, due dei quali possono essere usati per il driving di due LEDs (GPIO2 e GPIO3). Inoltre si ha un blocco che racchiude la logica di Start Up del dispositivo e che permette la generazione dei segnali di $\overline{RESET}$ e di interrupt $\overline{INT}$ e un'inerfaccia I2C che fornisce all'esterno le due linee SCLK e SDAT.
La tensione di alimentazione è fornita al TPS65721 tramite il pin d'ingresso AC. Al pin ISET, invece si collega un resistore(R5) di valore opportuno per settare la corrente di carica per la batteria. I due pins BAT devono essere collegati insieme ed entrambi collegati al terminale positivo della batteria al Litio/LiPo da

ricaricare. Il pin TS di thermal sensing è collegato ad un termistore NTC, collegato a sua volta al terminale di GND della batteria, ed usato per monitorare la temperatura della batteria stessa durante il suo funzionamento.
In particolare, è importante evidenziare che è stato scelto il dispositivo TPS65721 con package WQFN con 32 pins a cui corrisponde il seguente schema a blocchi funzionale:

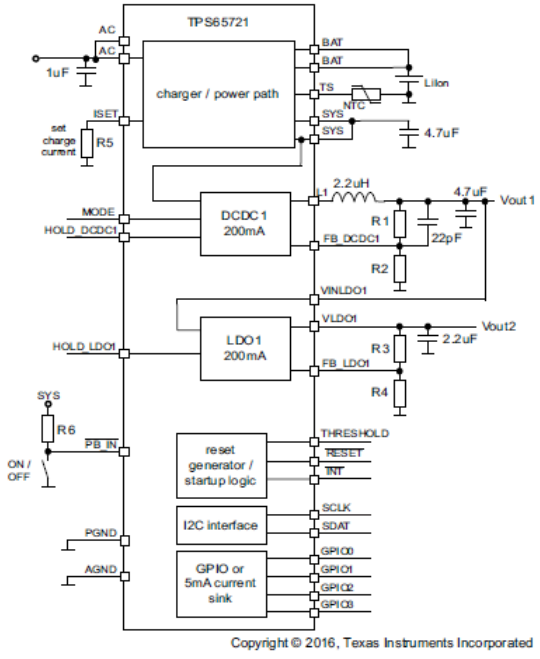


Figure 9. Functional Block Diagram for WQFN Version

Figura 3.3: TPS65721 schema funzionale, [93]

Un'altro pin importante è il $\overline{PBIN}$, a cui si può collegare un push-button, che può essere usato per attivare automaticamente il regolatore LDO e il convertitore DC-DC; i pins HOLD DCDC1 e HOLD LDO1 sono usati per mantenere attivi rispettivamente il convertitore DC-DC e il convertitore LDO.
I dispositivi della serie TPS6572x sono in grado di fornire due distinte tensioni d'uscita (Vout1 e Vout2), che possono essere usate per alimentare i componenti del sistema a cui sono collegati; contemporaneamente permettono la ricarica della batteria. Questo sistema garantisce la riduzione del numero di cicli di carica/scarica della batteria, e permette il funzionamento del sistema anche in assenza di una batteria esterna o in presenza di una batteria quasi completamente scarica.
Per quanto riguarda la gestione del Power Pathla corrente in ingresso al pin AC viene divisa in parte per ricaricare la batteria e in parte per fornire corrente ad un carico in uscita (System load) sul pin SYS. La priorità è data al pin SYS, pertanto

la corrente d'ingresso viene costantemente monitorata; se la somma della corrente di charging e della corrente verso il system load supera il valore massimo di corrente settato internamente, la corrente di charging è automaticamente ridotta.
Il valore di default per la corrente limite sul pin AC d'ingresso è pari a 500mA.
E' importante sottolineare che il processo di ricarica della batteria è suddiviso in tre fasi distinte:

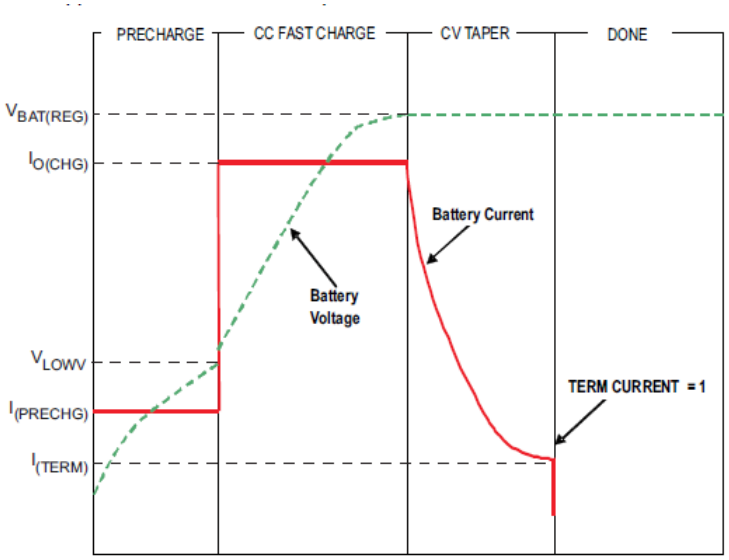


Figura 3.4: fasi ricarica batteria, [93]

Come mostrato in gura, 3.4, le tre fasi sono: precarica (PRECHARGE), carica veloce a corrente costante (CC FAST CHARGE) e carica a tensione costante (CV TAPER).
Nella fase di pre-carica, la batteria è ricaricata con una corrente costante pari a IPRECHG. Non appena la tensione della batteria supera il valore di soglia minima VLOW, si entra nella fase a carica veloce in cui il valore di corrente costante di ricarica risulta pari a ICHG. Dopodichè, quando la tensione della batteria raggiunge il livello della soglia massima pari a VBAT(REG) la corrente di carica comincia a ridursi no a raggiungere il valore minimo di ne carica ITERM, ossia la batteria ha raggiunto lo stato di "full charge".
Il valore della corrente di carica veloce è settato dal resistore connesso tra il pin ISET e GND(3.3), ed è dato dall'equazione :

$$ICHG = \frac{K_{ISET}}{R_{ISET}} \qquad (3.1)$$

Per determinare il valore di R_{ISET} sono stati adottati i seguenti passi di progetto.
Innanzitutto sono stati considerati i seguenti parametri del TPS65721, [93]:

® V_{AC} = *tensioned*^{0}*ingressosulpinAC* => $4.35V < V_{AC} < 28V$;

- V_{SYS} = *tensionesulpinSY S* => $2.2V < V_{SYS} < 5.6V$;
- I_{BAT} = *correntemediasulpinBAT* => $I_{BAT} = 300mA$;
- V *inLDO*1 = *tensioneiningressoall* ^{0}*LDO*1 => $1.8V < V$ *inLDO*1 $< V_{SYS}$;
- *ICHG* = *correntedicaricaveloce* => $40mA <$ *ICHG* $< 300mA$;

Tali valori sono stati utili nella scelta della batteria LiPo, [94], le cui caratteristiche principali sono :

- $V_{CUT-OFF} = 3.0V$;
- $V_{NOMINAL} = 3.7V$;
- $V_{CHARGE} = (4.2 \pm 0.03)V$;
- C = *capacitdellabatteria* => $C = 190mAh$;

Con correnti nelle diverse fasi di carica, pari a :

- $I_{PRECHG} = 0.5 * C = 95mA$;
- $I_{TERM} = 0.01 * C = 1.9mA$;
- $I_{FAST} = 1 * C = 190mA$;

Dunque il dispositivo TPS65721, per garantire la ricarica della batteria, deve fornire una $I_{CHG} = I_{FAST} = 190mA$. Per ottenere tale valore di I_{CHG} occorre considerare l'equazione (3.1)ll K_{ISET} è denito come il fattore di carica rapida ed assume i valori indicati nella seguente gura (U.M. [A Ω]), a seconda del valore impostato per i due bits $ICH_{SCL}0$ e $ICH_{SCL}1$ nel registro CHCONFIG1 :

K_{ISET}	Fast charge current factor	at 300 mA for ICH_SCL[1,0] = 11 (charge current scaling is 100% of ISET value)	–15%	450	15%	AΩ
		at 40 mA for ICH_SCL[1,0] = 11 (charge current scaling is 100% of ISET value)	–20%	450	20%	
		at 225 mA range for ICH_SCL[1,0] = 10 (charge current scaling is 75% of ISET value)	–15%	338	15%	
		at 30 mA for ICH_SCL[1,0] = 10 (charge current scaling is 75% of ISET value)	–20%	338	20%	
		at 150 mA for ICH_SCL[1,0] = 01 (charge current scaling is 50% of ISET value)	–10%	225	10%	
		at 20 mA for ICH_SCL[1,0] = 01 (charge current scaling is 50% of ISET value)	–15%	225	15%	
		at 75 mA for ICH_SCL[1,0] = 00 (charge current scaling is 25% of ISET value)	–10%	112	10%	
		at 10 mA for ICH_SCL[1,0] = 00 (charge current scaling is 25% of ISET value)	–20%	112	20%	

Figura 3.5: valori di K_{ISET} , [93]

ICH_SCL1	ICH_SCL0	ISET = %*ICHG
0	0	25%
0	1	50%
1	0	75%
1	1	100%

Figura 3.6: bits $ICH_{SCL[1:0]}$

Basandosi sui valori indicati nell'ultima gura, 3.6, è facile ricavare il range di possibili valori per R_{ISET} (considerando per esempio $ICH_{SCL1} = ICH_{SCL0} = {}^{0}1^{0}$) e dunque $I_{SET} = I_{CHG}$ e $K_{ISET} = 450$:

$$I_{CHG} = 300mA => R_{ISET} = \frac{K_{ISET}}{I_{CHG}} = 1.5k\Omega \tag{3.2}$$

$$I_{CHG} = 40mA => R_{ISET} = 11.25k\Omega \tag{3.3}$$

Poichè la batteria scelta necessita di una $I_{CHG} = I_{FAST} = 190mA$, è stato realizzato un'opportuno script Matlab in modo da ottenere quattro distinti graci di I_{CHG} in funzione di R_{ISET}, in base alla scelta dei bits di congurazione $ICH_{SCL}[1:0]$. Analizzando i graci ottenuti, è stato scelto il caso con $ICH_{SCL}[1:0] =$ "00" e quindi $I_{SET} = 0.25 * I_{CHG}$: con $I_{CHG} = 190mA$ la R_{ISET} corrispondente assume un valore di circa 2393. Considerando i resistori della serie E12 è stata scelta $R_5 = R_{ISET} = 2.7k\Omega$, a cui corrisponde un valore di $I_{CHG} \approx 170.26mA$.
Un valore più basso rispetto ai 190 mA, ciò implica un tempo di ricarica leggermente superiore, evitando tuttavia una sovracorrente che danneggerebbe la batteria stessa. Dopodichè, sempre seguendo quanto riportato sul datasheet, [93], sono stati ssati i valori di R1 ed R2 per ottenere 3.3V all'uscita del convertitore DCDC1 (Vout1), rispettivamente pari a 680 kΩ e 150 kΩ. Mentre per ottenere all'uscita del convertitore LDO (Vout2) un valore di 1.8 V, sono stati scelti R3 pari a 300 kΩ ed R4 pari a 240 kΩ. Inne, il termistore NTC è stato scelto pari a 10 k Ω, con il parametro beta(β) pari a 3380 K.
Nella seguente gura è rappresentata la schematic view del componente, realizzata in Altium Designer :

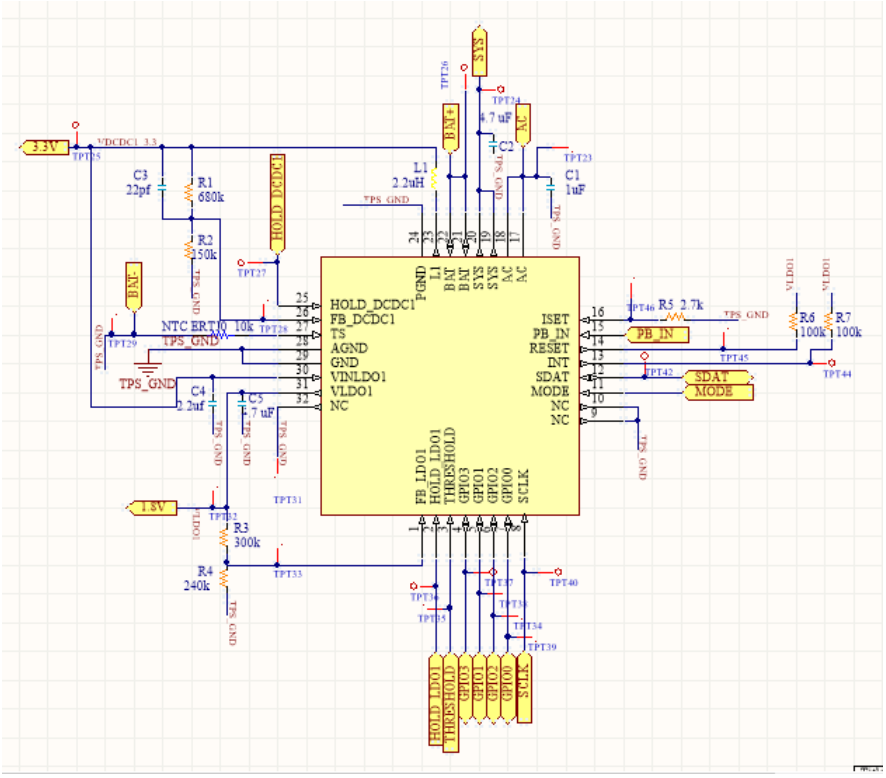


Figura 3.7: TPS65721 schematic view

3.2.1 Scelta dei componenti

Il progetto della "mini-PCB" è stato, pertanto, sviluppato tenendo in considerazione il vincolo fondamentale dell'area. In tale ottica, di conseguenza, i sensori e i componenti attivi (es: memorie, microcontrollore etc...) sono stati scelti in base al package di dimensioni minori. Allo stesso modo, i componenti passivi (es: condensatori, resistenze, induttori etc...) sono stati selezionati con dimensioni comprese tra gli 0402 e i 0603 pollici.
Il secondo vincolo importante nella scelta dei componenti attivi è stata la tensione di alimentazione. Infatti, il sistema è stato progettato per poter essere alimentato da due sorgenti diverse: una batteria LiPo [94], con tensione nominale di 3.7 V e tensione di cut-o di 3.0V ; tramite la porta micro-USB e pertanto come previsto dallo standard USB si ha una tensione di 5.0V.
Dunque, sono stati selezionati dispositivi che potessero funzionare con tensioni di alimentazione comprese fra 3.0V e 5.0V.

Sensori

I sensori che sono stati scelti sono i seguenti:

- LSM6DSO iNemo : accelerometro 3D e giroscopio 3D della ST Microelectronics, [79];
- LIS3MDL : magnetometro triassiale della ST Microelectronics, [80];
- sensore GSR della Grove, [96];

E' importante sottilineare, che per la PCB miniaturizzata sono stati scelti i singoli chip integrati relativi ai sensori sopracitati; mentre nel Capitolo 2 i sensori digitali descritti si trovavano all'interno di schede di valutazione già pre-realizzate: l'LIS3MDL all'interno della STEVAL-MKI137V1 ; l'LSM6DS0 all'interno della STEVAL-MKI161V1 ; il sensore pulsimetro MAX30105 all'interno dell'omonima scheda di valutazione prodotta da Sparkfun.
Tutti gli schematici che verranno mostrati in seguito sono stati realizzati con il software di design delle PCB Altium Designer.

LSM6DSO iNemo L'LSM6DS0 iNemo è un system-in-package (SiP) che contiene sia un'accelerometro 3D sia un giroscopio 3D, entrambi realizzati con la tecnologia MEMS. Inoltre, al suo interno sono integrati anche un sensore di temperatura e una memoria di tipo FIFO.
I diversi elementi di sensing sono prodotti usando specici processi di micromachining, mentre le interfacce IC sono sviluppate usando la tecnologia CMOS.
Trattandosi di un sensore digitale, i dati d'uscita sono trasmessi in modo seriale sfruttando un'interfaccia di tipo SPI o I^2C, a seconda di come viene programmato.
Il package è di tipo LGA a 16 pins con dimensioni "3x3x0.86" mm.

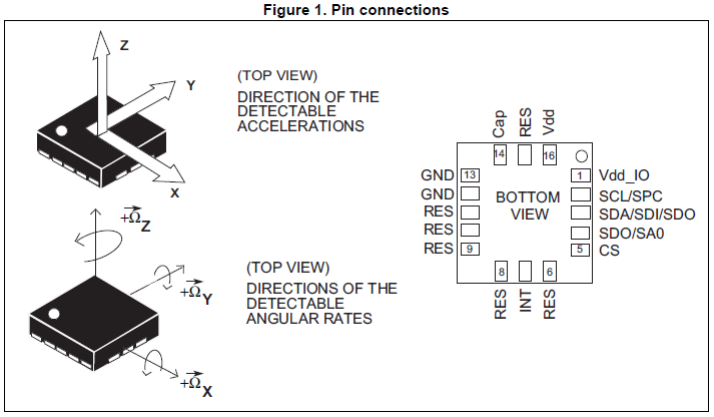


Figura 3.8: LSM6DS0 package, [79]

Per quanto riguarda la tensione di alimentazione, Vdd è compresa tra 1.71 V e 3.6 V ; è indicata, inoltre una seconda tensione di alimentazione V_{dd_IO} compresa tra

1.71 V e Vdd + 0.1 V, usata solo per le misurazioni delle grandezze d'ingresso (I) e la trasmissione dei dati d'uscita (O). In quest'applicazione i due pins sono stati collegati insieme, poichè si vuole gestire un'unica linea d'alimentazione. Nella gura seguente sono state riportate le Caratteristiche elettriche del sensore:

@ Vdd = 2.2 V, T = 25 °C unless otherwise noted

Table 4. Electrical characteristics

Symbol	Parameter	Test conditions	Min.	Typ.(1)	Max.	Unit
Vdd	Supply voltage		1.71		3.6	V
Vdd_IO	Power supply for I/O		1.71		Vdd + 0.1	V
LA_Idd	Accelerometer current consumption in normal mode	ODR = 10 Hz		60		µA
		ODR = 50 Hz		160		
		ODR ≥ 119 Hz		330		
G_Idd	Gyroscope current consumption in normal mode			4.0		mA
Top	Operating temperature range		-40		+85	°C
Trise	Time for power supply rising(2)		0.01		100	ms
Twait	Time delay between Vdd_IO and Vdd (2)		0		10	ms

Figura 3.9: Caratteristiche elettriche LSM6DS0, [79]

Per comprendere, invece, le Caratteristiche meccaniche di questo dispositivo inerziale occorre denire la seguente terminologia: Measurement range, Sensitivity, Zero-g level, Zero-rate level e Output Data Rate.
Per Measurement range, o range di misura appunto, s'intende il più grande valore di una data grandezza, in questo caso accelerazione lineare e accelerazione angolare, che può essere misurato e rappresentato come un preciso valore d'uscita. Per esempio se si considera un'accelerometro $\pm 3g$ (ossia il FSR (Full Scale Range)), la sua uscita sarà lineare sino ad accelerazioni in modulo pari a 3g (ossia il FS (Full Scale)). Se, invece, fosse accelerato a 4g, dunque al di fuori del suo range di misura, l'uscita potrebbe deviare dalla linearità. Da notare, che ciò non indica il punto di rottura del sensore, che è da ricercare tra gli Absolute Maximum Ratings e non nel range di misura.
La Sensitivity è denito come un parametro dierenziale, come il rapporto tra la variazione dell'ingresso e la variazione del segnale d'uscita. Naturalmente, si vuole tale valore il più costante possibile al variare dell'ingresso, in modo da avere una curva "output-input" del sensore il più lineare possibile. Poichè si tratta di un dispositivo inerziale con uscita digitale, le due sensitivity sono espresse in $\frac{mg}{LSB}$ e in $\frac{mdps}{LSB}$ rispettivamente per l'accelerazione lineare e l'accelerazione angolare (dps = degree per second). Occorre considerare che la sensitivity cambia in funzione della temperatura del dispositivo.
Come detto, nel caso ideale la relazione tra uscita ed ingresso del sensore è lineare se la sensitivity è costante. Tuttavia, spesso bisogna considerare numerosi eetti di non-linearità. Per non-linearità s'intende la misura della deviazione da una

sensitivity costante ed è espressa come percentuale del "Full Scale Range (%FSR)" o percentuale del $\pm$ "Full Scale (FS)", dove tipicamente $FSR = FS + FS$.
La sensitivity non è da confondere con la Risoluzione del sensore, denita come la più piccola variazione della grandezza d'ingresso che può essere misurata, dunque la minima variazione dell'ingresso che porta ad una variazione apprezzabile dell'uscita.
Lo Zero-g levelspecica il valore d'uscita dell'accelerometro, quando si ha un'ingresso nullo (dunque nessuna accelerazione lineare). Tipicamente nei sensori digitali viene espresso in LSB. Lo Zero-g levelè denito una volta ssata la tensione d'alimentazione; in particolare possono esserne specicati diversi aspetti:

- Zero-g voltage: espresso in $[V]$, denisce il range di tensioni, che ci si potrebbe aspettare in uscita per accelerazioni $= 0g$;
- Outpur deviation from ideal anche noto come Initial bias error, è denito alla temperatura di $25^{\circ}C$, o in termini di errore di accelerazione (g) o come errore sul segnale d'uscita: dunque in $[mV]$ per i sensori analogici e in LSB per i sensori digitali.
- Zero-g oset vs temperature o Bias temperature coecient, è espresso in $[\frac{mg}{^{\circ}C}]$ e denisce di quanto l'uscita, con ingresso nullo, varia per ogni grado $^{\circ}C$ di variazione della temperatura;
- Bias voltage sensitivity, denisce la variazione dello Zero-g levelrispetto alla variazione della tensione di alimentazione;

Lo Zero-rate levelesprime lo stesso paramtro, ma riferito al giroscopio (dunque all'accelerazione angolare) e denisce, infatti, il valore dell'uscita quando nessuna velocità angolare è misurata all'ingresso. Di solito lo Zero-rate level di un sensore MEMS è il risultato o di stress meccanici applicati al sensore o può essere dovuto al montaggio su una PCB.
Inne, per Output Data Rate (ODR), s'intende la massima frequenza a cui i dati sono campionati senza che si verichi il fenomeno dell'aliasing. La Schematic view, di questo come per tutti gli altri componenti, è stata realizzata con il software Altium Designer, basandosi sull'Application circuit indicato nel datasheet.

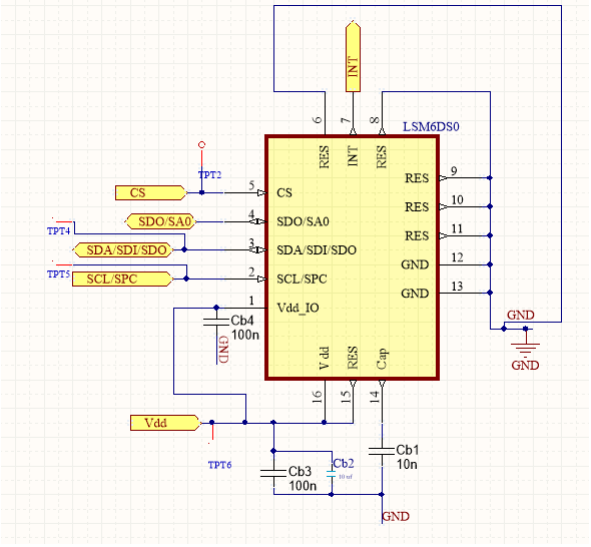


Figura 3.10: LSM6DS0 schematic

LIS3MDL Il LIS3MDL è un magnetometro triassiale prodotto dalla ST Microelectronics, ad elevate prestazioni. E' fabbricato in un package di tipo LGA a 12 pins:

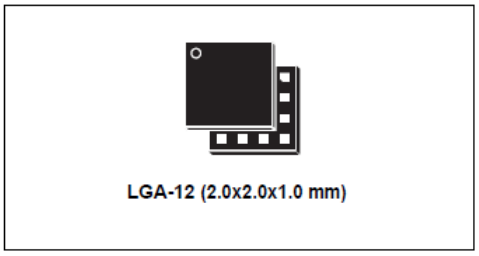


Figura 3.11: LIS3MDL package, [80]

Questo dispositivo può essere alimentato con tensioni nel range tra 1.9 V e 3.6 V. Il numero di bits dei dati in uscita è rappresentato dalla risoluzione dell'ADC ed è pari a 16. Un'altra caratteristica interessante è data dall'ampia scelta di fondo scala (FS) selezionabili: $\pm 4, \pm 8, \pm 12, \pm 16$[*gauss*].
Come detto nel capitolo precedente, le principali applicazioni sono quelle di rilevatore del campo magnetico terrestre e di bussola.
Lo schematico di questo componente è stato realizzato basandosi sul tipico Application circuit mostrato nel datasheet, [80]:

Figure 5. LIS3MDL electrical connections

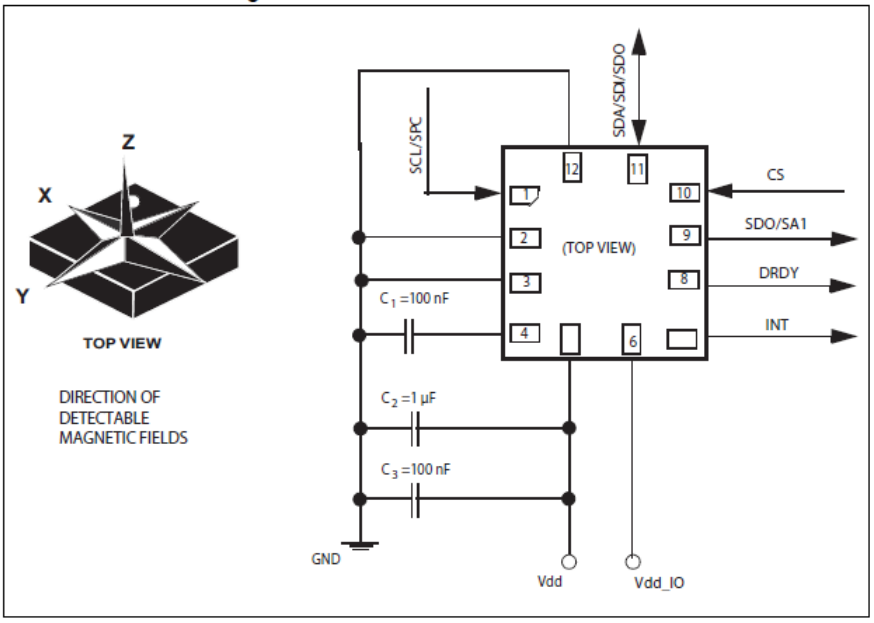


Figura 3.12: LIS3MDL : application circuit

Come mostrato in gura 3.12, ai pins del magnetometro sono connessi condensatori di valori diversi: il condensatore C1 da 100 nf connesso tra il pin 4 e GND; sulla linea di alimentazione (Vdd) è necessario l'inserimento di un condensatore ceramico, in alta frequenza, di disaccoppiamento (C3) del valore di 100 nf, da posizionare il più vicino possibile al pin Vdd stessoInoltre, è importante inserire un condensatore elettrolitico in bassa frequenza C2 da 1 uf.
I pins relativi al protocollo di comunicazione seriale I2C/SPI (dunque pin 1 e pin 11) permettono di accedere ai dati digitali che si riferiscono al campo magnetico misurato. Tuttavia, anchè si possano usare le comunicazioni I2C ed SPI, il pin SDO/SA1 dev'essere connesso a V_{dd_O} o a GND; ciò cambierà inoltre l'indirizzo stesso del sensore.
Inne, nella gura seguente è mostrato lo schematico Altium del circuito appena descritto:

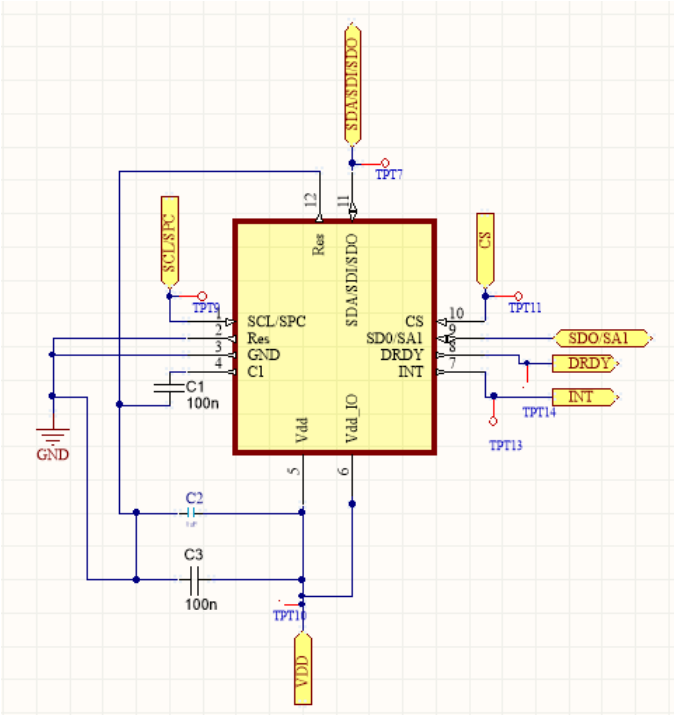


Figura 3.13: LIS3MDL schematico Altium

Sensore GSR Il sensore GSR, come detto, è usato per la misurazione della conduttanza/resistenza della pelle umana posizionandolo sulle estremità del corpo come mani e piedi.
Come è stato descritto nel capitolo precedente, è stato scelto il sensore GSR prodotto dalla Seed Studio Electronicsll datasheet del sensore presenta il seguente circuito di condizionamento:

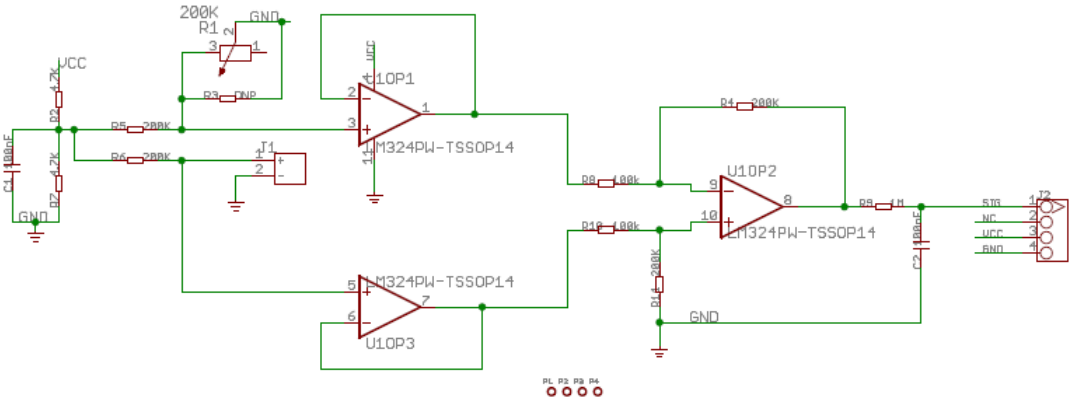


Figura 3.14: Sensore GSR : circuito condizionamento, [96]

Si tratta di un circuito di condizionamento abbastanza complesso, infatti svolgere la misurazione di uno stato emozionale richiede un processo accurato e complesso. Nel connettore femmina J1 andranno inseriti i li collegati agli alloggiamenti in cui le dita dovranno essere inlate per l'esecuzione delle misurazioni. Quindi, l'uscita positiva (+) del connettore J1 va all'ingresso di due amplicatori operazionali in congurazione voltage follower :in questo modo, grazie al valore elevato d'impedenza d'ingresso e il ridotto valore d'impedenza d'uscita, si riesce ad ottenere un'ottimo disaccoppiamento d'impedenza.
Dopodichè, le uscite dei due voltage followers sono messe in ingresso ad un'operazionale in congurazione dierenziale, che amplica appunto la dierenza dei suoi ingressi di un fattore pari al suo guadagno.
L'intero sistema è alimentato da una tensione Vcc che può variare nel range tra 3.3 V e 5.0 V. Un'importante parametro del sensore, ossia la sua Sensitivity può essere regolata attraverso il potenziometro R1 da 200 kΩ, in alto a sinistra nella gura 3.14.
Il connettore J2 fornisce quattro segnali in output: l'uscita analogica in tensione del GSR; il valore di Vcc; la connessione di GND e un pin NC che non dev'essere connesso a nulla.
A livello di chip sico, i tre amplicatori indicati fanno parte del quadruple amplier integrato LM324 della Texas Instruments, con package di tipo TSSOP a 14 pins (5.00 x 4.40 mm).Nella scheda miniaturizzata è stato mantenuto lo stesso circuito di condizionamento, ma al posto dell'LM324 è stato scelto l'amplicatore quadruplo MCP6004 della Microchip Technology, [95]:

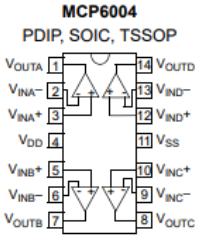


Figura 3.15: MCP6004 package, [95]

E' stato adottato tale chip per il range della tensione di alimentazione, compreso tra 1.8 V e 6.0 V, più adatto alla nostra applicazione. Infatti, permette di scendere sino ad un valore minimo di 1.8 V, nel caso in cui si vogliano ridurre ulteriormente i valori delle potenze dissipate dai componentiPer ora tutti i sensori sono stati alimentati con tensione di 3.3 V.

Inne, nella seguente gura è stato riportato lo schematico del circuito di condizionamento completo del GSR:

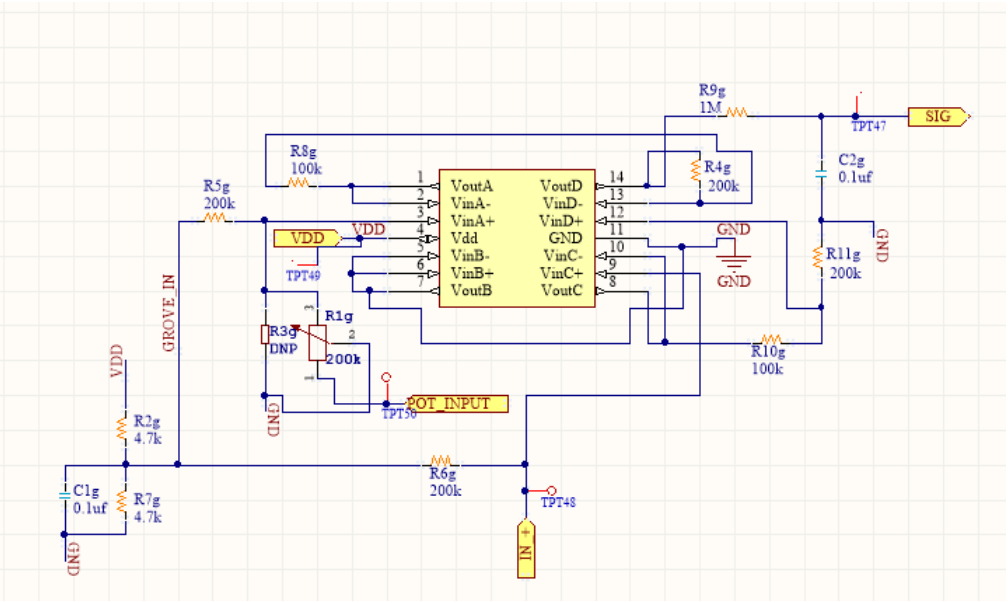


Figura 3.16: Schematico circuito condizionamento del GSR

Sensori di forza e sensori di bending I sensori di forza sono di tipo FSR(Force Sensing Resistor), il cui principio di funzionamento stabilisce che all'aumentare della forza applicata si riduce la resistenza del sensore. Sono stati scelti, come descritto nei capitoli precedenti, i sensori A301 della Flexiforce. Tuttavia, anche in questo caso, per ottenere un valore d'uscita che possa essere misurato, occorre inserire un'opportuno circuito di condizionamento all'uscita dei sensori di forza. Dopo aver consultato lo User Manual, [98], dei sensori FSR Flexiforce e aver completato una ricerca delle possibili soluzioni presenti in letteratura, i due seguenti sono i circuiti di condizionamento più comunemente usati:

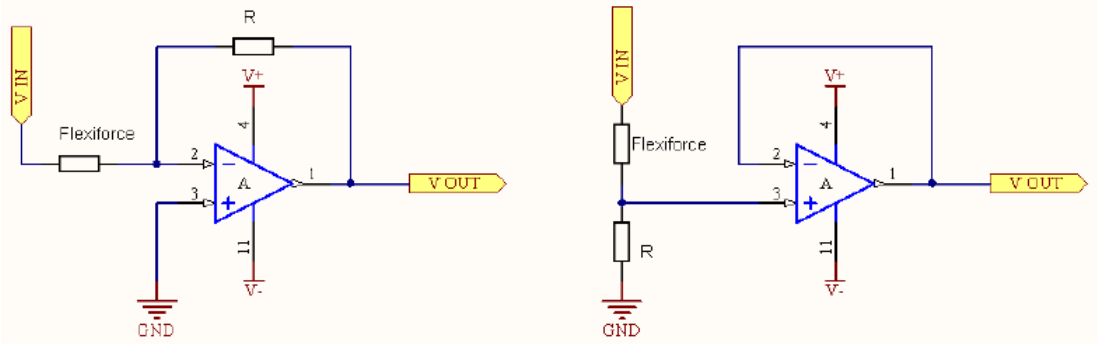


Figura 3.17: Circuiti di condizionamento per i sensori di forza, [77]

E' stata scelta la congurazione a destra, nell'immagine 3.17, ossia il partitore di tensione seguito da un voltage follower, in modo tale che scegliendo una tensione d'alimentazione Vdd positiva (e dunque un range single rail da GND a Vdd), si ottenga una tensione d'uscita Vout ancora positivaPertanto, si evita d'incrementare la complessità del circuito e si mantiene limitata l'area occupata. Inoltre, in questo modo l'uscita in tensione è piuttosto stabile, in quanto si ha un buon disaccoppiamento tra ingresso e uscita, grazie all'elevato valore d'impedenza d'ingresso della congurazione non-invertente.
La relazione ingresso-uscita di questo circuito è data dalla seguente espressione:

$$V_{out} = V_{in} * \frac{R}{R_{flex} + R}$$

A questo punto, sono stati deniti i valori numerici dei diversi elementi del circuito. Come già stabilito, la tensione di alimentazione dei sensori è pari a 3.3 V. Dopodichè, è stato eseguito un processo di calibrazione applicando delle masse [Kg] note al sensore e misurando la resistenza corrispondente [Ohm]. Per Calibrazione, [98], s'intende appunto il metodo che permette di determinare la relazione tra l'uscita elettrica del sensore ed un'eettiva unità di misura ingegneristica, come appunto i [Kg] o i [N]. In generale, si applicano masse note e dunque speciche forze-peso $[Kg * m/(s^2)]$ in modo da approssimare il più possibile, il range di misura dell'applicazione in cui verrà usato.
Come indicato nel "Manuale Utente", [98], nei sensori Flexiforce della Tekscan, la relazione tra resistenza (output) e Forza (input) è altamente non-lineare; pertanto, spesso è conveniente lavorare in termini di conduttanza $1/R$ che, al contrario, presenta un'andamento piuttosto lineare in funzione della forza applicata:

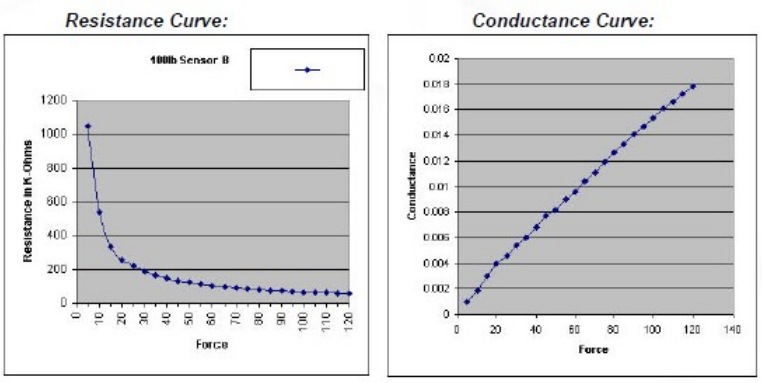


Figura 3.18: Graci resistenza e conduttanza, [98]

Nel nostro caso, per il sensore A301 analizzato, la relaizione non-lineare tra massa d'ingresso e resistenza d'uscita è pari a:

$$R_{flex} = 2.36 * 10^{(} * massa - 2.02)$$

Dunque, sono stati sostituiti i valori di Vdd ed R_{flex} nell'espressione della tensione d'uscita di V_{out}; dopodichè è stato plottato un graco di V_{out} (asse Y) in funzione della Massa (asse X), facendo variare il valore di resistenza R del circuito di condizionamento:

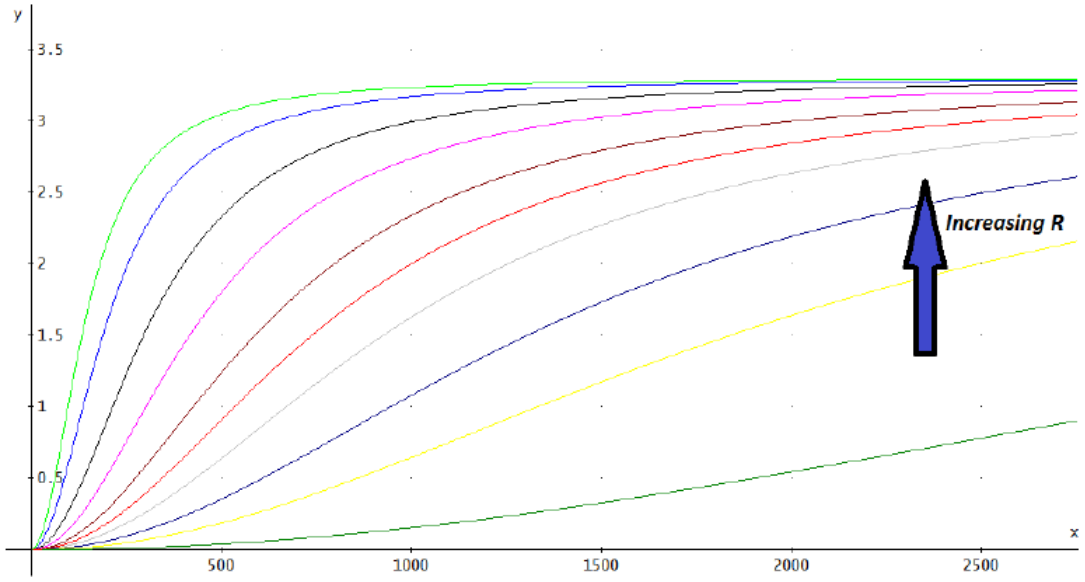


Figura 3.19: Graco $V_{out} - massa$, [77]

Dalla gura, 3.19, è facile notare come tutte le curve presentano un'andamento asintotico sino a 3.3 V. Al crescere del valore della resistenza R, l'uscita V_{out} satura al valore di Vdd per valori di forza applicata sempre minore; al contrario, il

range di forze applicabili cresce al ridursi del valore di resistenza R. Quindi, anche il valore di R risulta essere un'importante parametro per stabilire se tale sensore può essere usato per una data applicazione o meno.
Per la nostra applicazione, del Wired-glove, il range scelto lungo l'asse X è tra 0 e 2 Kg; dopodichè, è stata scelta la curva che ricopre il maggior numero di valori tra 0 e 3.3 V (lungo asse Y) e che sia il più lineare possibile.
Dunque, sono state eliminate le prime quattro curve partendo dall'alto nel graco 3.19, poichè saturano in prossimità dei 500g; inoltre, sono state eliminate le prime tre curve partendo dal basso poichè ricoprono pochi valori lungo Y.
Di conseguenza, sono state considerate soltanto le tre curve centrali, in particolare è stata scelta la curva rossa. Il valore della resistenza R corrispondente è di 31.6 k Ω.
Per quanto riguarda, invece, i sensori di bending, gli steps progettuali seguiti sono molto simili. Tuttavia, è stato un po' più complicato plottare il graco relativo al variare della resistenza del sensore in funzione dell'input applicato. Ciò è dovuto al fatto che la resistenza del sensore di bending varia in funzione di due parametri: primo è il raggio di bending, più piccolo è il raggio, maggiore sarà la variazione di resistenza; il secondo consiste nella deessione angolare, ssato il raggio, più il sensore è piegato, maggiore sarà la variazione.
Anche per il sensore di bending, come circuito di condizionamento, è stato usato un semplice amplicatore operazionale in congurazione voltage follower, preceduto da un partitore di tensione; dunque lo stesso usato per i sensori di forza:

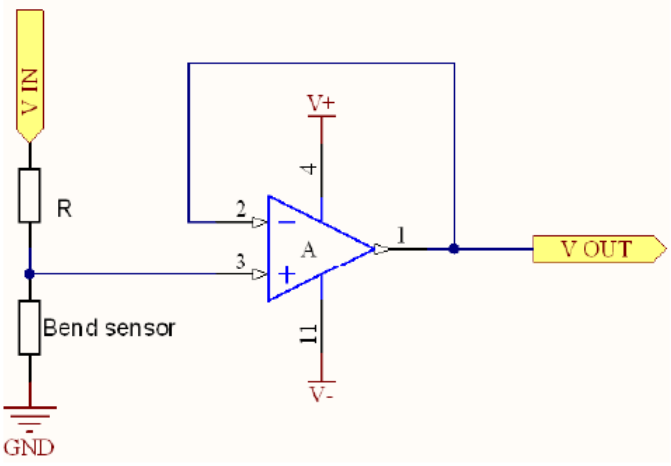


Figura 3.20: circuito condizionamento sensore di bending, [77]

L'unica dierenza è rappresentata dal diverso posizionamento del sensore di bending stesso, connesso tra il nodo + dell'amplicatore e il GND.
Di conseguenza, la relazione ingresso-uscita risulta essere:

$$V_{out} = V_{in} * \frac{R_{bend}}{R_{bend} + R}$$

Quindi, al crescere della deessione, cresce la resistenza del sensore e dunque anche il valore di tensione d'uscita V_{out}.
Considerato che $V_{in} = V\,dd = 3.3V$, il passo successivo è stato determinare il valore di R che massimizza il range di deessione del sensore. Il datasheet del sensore, [35], indica che R può variare tra 10 kΩ e 10 MΩ.
Poichè, il sensore di bending sarà collocato sulle articolazioni PIP (di indice, medio, anulare e mignolo), per la nostra applicazione il range di essione sarà tra 0 e 90 gradi.
Dopo aver eseguito il testing del sensore, è stato scelto 3.3 MΩ come valore nale per R (scelto dalla serie E24), che permette di ottenere la miglior caratteristica d'uscita in termini di range della pressione d'ingresso e della stabilità del segnale di tensione d'uscita.
Come menzionato in precedenza, risultano necessari sia quattro sensori di forza sia quattro sensori di bending da posizionare su altrettante dita della mano. Per rispettare il vincolo dell'area, è stato scelto un'unico integrato, ossia l'amplicatore quadruplo MCP6004,[95]: in tal modo è possibile collegare i quattro sensori con i rispettivi circuiti di condizionamento ad un'unico chip ed avere disponibili le rispettive quattro uscite in tensione.
Questa procedura, è stata adottata sia per i sensori di bending sia per i sensori di forza; in tal modo si hanno soltanto due chip integrati al posto di otto operazionali singoli, che avrebbero occupato un'area circuitale superiore.
Gli schematici relativi ai sensori di forza e ai sensori di bending sono stati riportati nelle seguenti gure:

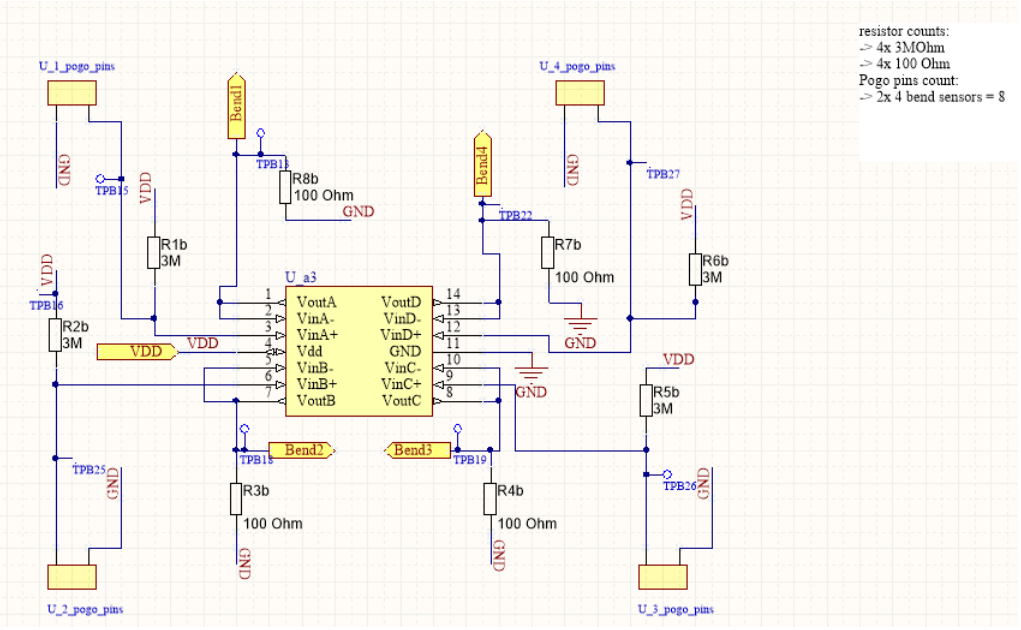


Figura 3.21: Schematico sensori di bending

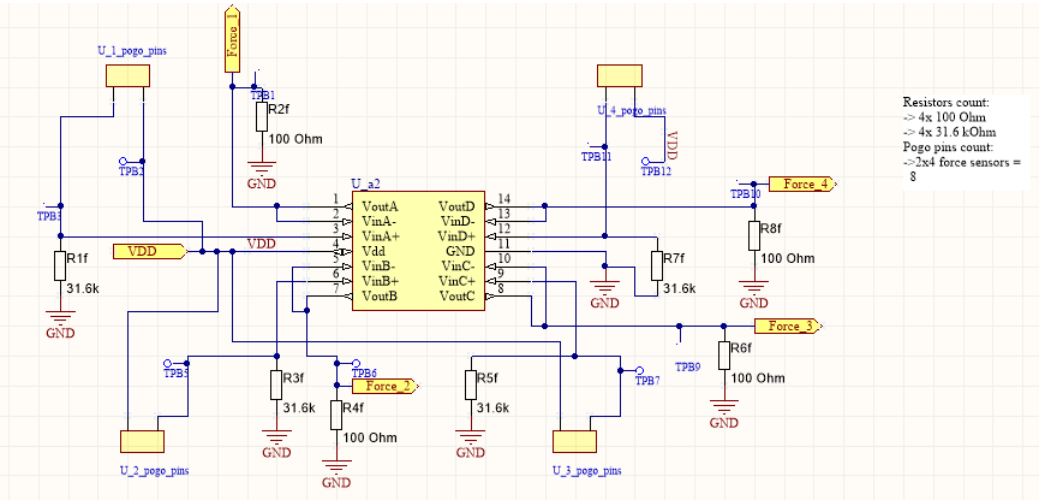


Figura 3.22: Schematico sensori di forza

Microcontrollore Il microntrollore scelto è l'nRF52832 della Nordic Semiconductor.Si tratta di un SoC (System on Chip) di tipo Ultra-low power, dotato di un transceiver integrato funzionante alla frequenza di 2.4 GHz, che permette l'utilizzo del chip in applicazioni wireless basate sugli standards Bluetooth low energy ed ANT. Inoltre, è dotato dell'hardware necessario a supportare lo standard Bluetooth 5.0.
L'nrF52832 è costruito attorno ad una CPU da 32 bit Cortex-M4F della ARM, con memoria RAM da 512 KB + 64 KB. Questo microprocessore supporta istruzioni DSP, è dotato di un'unità oating point (FPU), un blocco MAC (multiply and accumulate) a singolo ciclo e di un divisore hardware per svolgere ogni tipo di operazione matematica complessa nel modo più eciente possibile, in termini energetici.
Come detto l'nrF52832 presenta ridotti consumi di potenza, anche grazie ad un range di tensione di alimentazione che varia tra 1.7 V e 3.6 V. Inoltre, è supportato un NFC A-tag per le comunicazioni wireless a distanze ridotte e che necessitano di bassi data rate di comunicazione dati.Da sottolineare che l'nrF52832 supporta anche l's132 SOftDevice, ossia uno stack protocollo basato sul Bluetooth 5.0, che permette la connessione sino a 20 dispositivi contemporaneamente, [101].
Nordic Semiconductor lo fornisce sotto forma di le binario pre-qualied, dunque che non dipende dallo sviluppo della specica applicazione.
Le opzioni relative al package sono duepackage QFN da 48 pins (6x6 mm) e un WL-CSP (ultra compact wafer-level chip scale package) da 3.0x3.2 mm.
Nella scheda prototipo è stato adottato, come già citato nel Capitolo 2, il microcontrollore Aconno ACN52832 basato sull'nrF52832 :

Figura 3.23: ACN52832 della Aconno, [101]

Si tratta di un dispositivo adatto per le applicazioni IoT, disponendo di un'antenna RF integrata (parte superiore della Top View in gura 3.23). Inoltre, è dotato di un connettore di tipo Tag-Connect, il connettore nella Back View dell'immagine 3.23, per una rapida ed eciente programmazione del chip.Le altre caratteristiche importanti di questo chip sono : la possibilità di pilotare sino a 32 pins di GPIO;

sono integrati, oltre all'antenna RF, anche due oscillatori basati su cristalli di quarzo, da 32 MHz e da 32.768 KHz per la generazione dei segnali di clock, ed un LED RGB; un'interfaccia di Debug SWD (Serial Wire Debugging) per il testing e il debugging del micro nel caso non si disponga di un Tag connector ; inne, un'interfaccia seriale permette la trasmissione/ricezione dati attraverso i protocolli I2C, SPI ed UART. E' possibile, inoltre, collegare un'antenna NFC esterna.
Lo schema a blocchi di quanto descritto nora, è mostrato nella gura seguente:

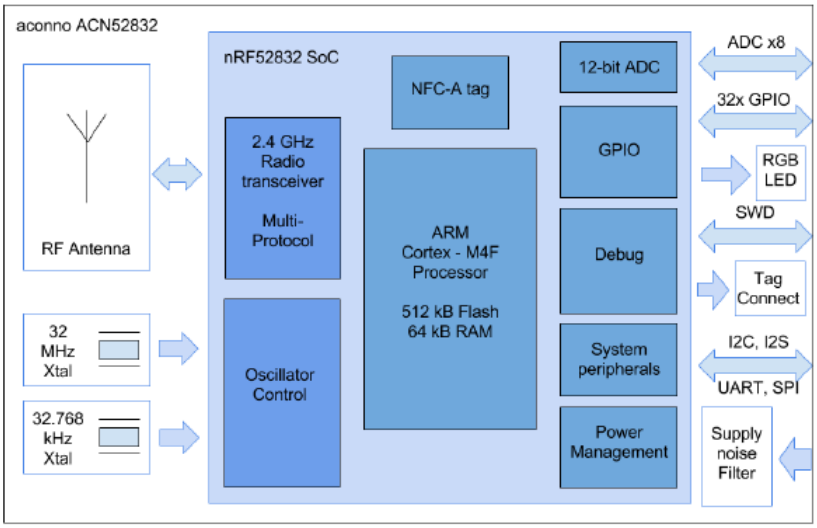


Figura 3.24: Schema a blocchi del chip ACN52832, [101]

Il modulo Aconno dispone di un totale di 35 pins, equispaziati con un pitch di 1.4 mm, e può essere facilmente saldato su una PCB sia con montaggio di tipo through hole, sia con montaggio superciale SMD :

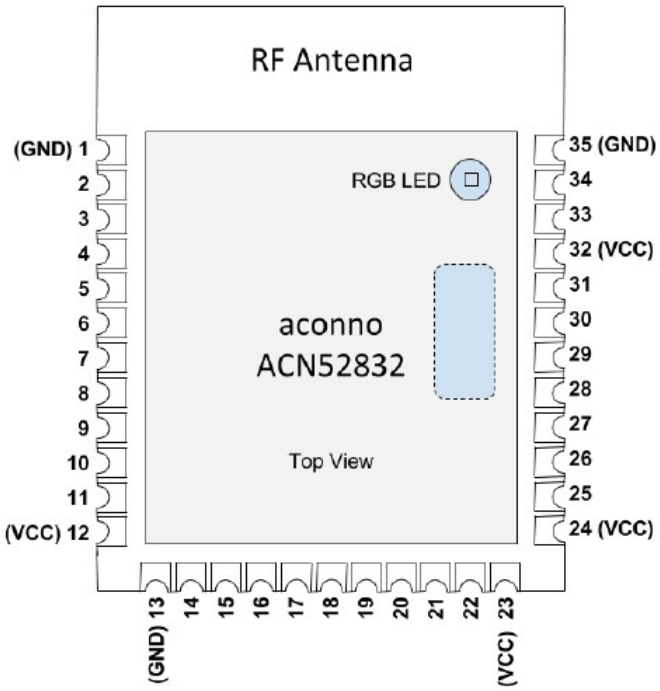


Figura 3.25: Aconno Top View, [101]

Per quanto riguarda, invece, la realizzazione della scheda miniaturizzata, è stato adottato un diverso chip integrato, basato sempre sull'nrF52832 e prodotto dalla Taiyo Yuden. Infatti, le dimensioni del modulo ble Aconno 20.3x25x3 mm, sono risultate eccessivamente grandi, di conseguenza il chip avrebbe occupato quasi la metà dell'intera supercie della PCB, non lasciando spazio per i restanti componenti. Tra i moduli bluetooth disponibili, solo tre di questi garantivano la compatibilità con il Bluetooth 5.0 : l'EYSHSNZWZ, l'EYSHJNZWZ e l'EYSHJNZWZ.

	EYSHSNZWZ	EYSHJNZWZ	EYSHCNZWZ
Dimension (mm)	3.25 x 8.55 x 0.85	5.1 x 11.3 x 1.3	9.6 x 12.9 x 2.0
Chip	Nordic nRF52832 with S132 V5.0.0 SoftDevice ARM Cortex M4F, Flash 512KB, RAM 64KB		
GPIO Pin	15	15	30
32kHz Crystal	External	External	Internal
LC for DCDC	External	External	Internal
Common Specifications	Embedded Antenna, FCC, ISED, MIC Certified, Bluetooth Logo Qualified, Bluetooth Version v5.0		

Figura 3.26: 3 moduli ble a confronto, [102]

Tra questi è stata scelta la versione con l'ultra smalpackage, ossia l'EYSHSNZWZ, che presenta l'area più piccola e costituisce pertanto una caratteristica fondamentale per il nostro progetto di wired-glove:

Figura 3.27: EYSHSNZWZ Taiyo Yuden, [102]

Nella gura 3.27, il modulo ble considerato è quello a sinistra; come è facile notare le sue dimensioni sono notevolmente inferiori rispetto ad una generica Micro SD, a destra in gura 3.27, e ridotte di circa il 48 per cento rispetto al modulo Taiyo più piccolo, in centro nella gura 3.27.
Il package è di tipo WLCSP. L'unico svantaggio di questo modulo rispetto all'Aconno, è rappresentato dal minore numero di pins di GPIO pari a 15 :

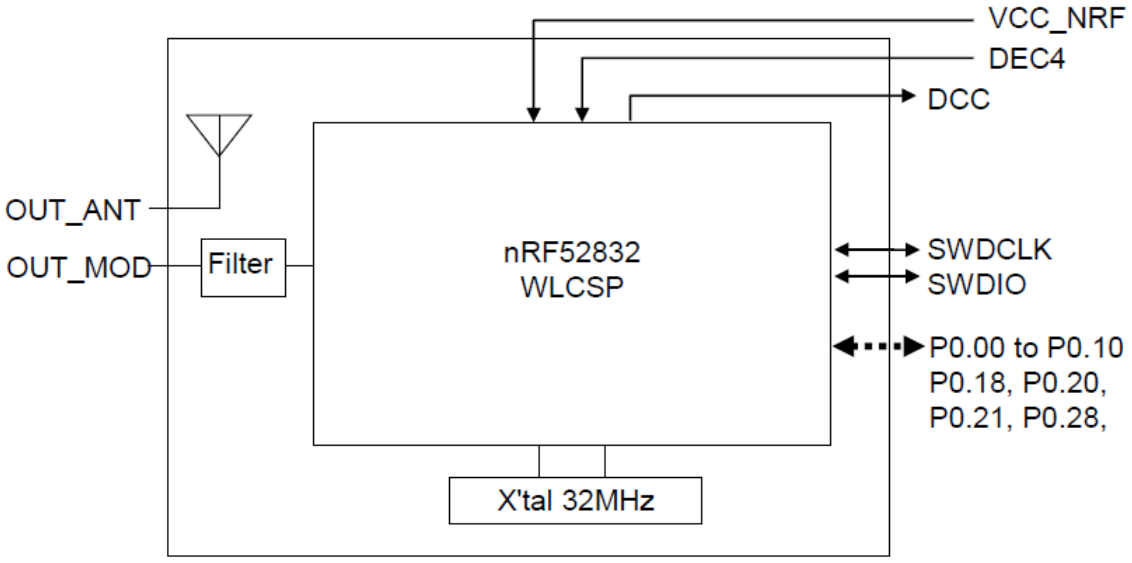


Figura 3.28: Schema a blocchi dell' EYSHSNZWZ, [103]

Per la realizzazione dello schematico, sono state considerate le due gure seguenti, in cui è mostrato come connettere il cristallo esterno da 32.768 kHz, e i pins DCC e DEC4 mediante due induttori per il corretto setup rispettivamente dell'LDO e del convertitore DC-DC interni al modulo Taiyo:

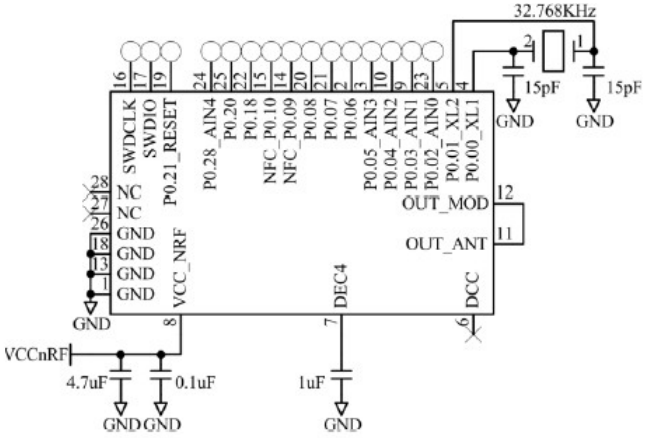


Figura 3.29: LDO setup Taiyo, [103]

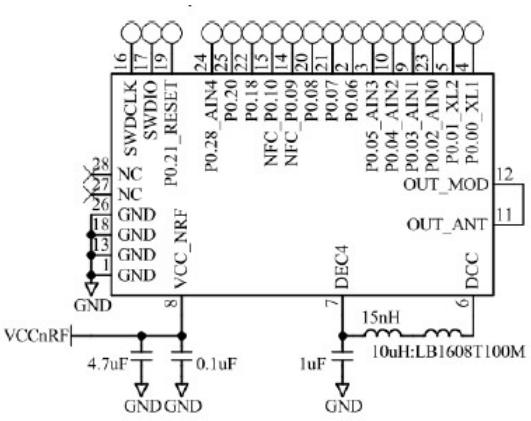


Figura 3.30: DC DC setup Taiyo, [103]

Inne, è riportato lo schematico, realizzato in ALtium, del chip bluetooth Taiyo :

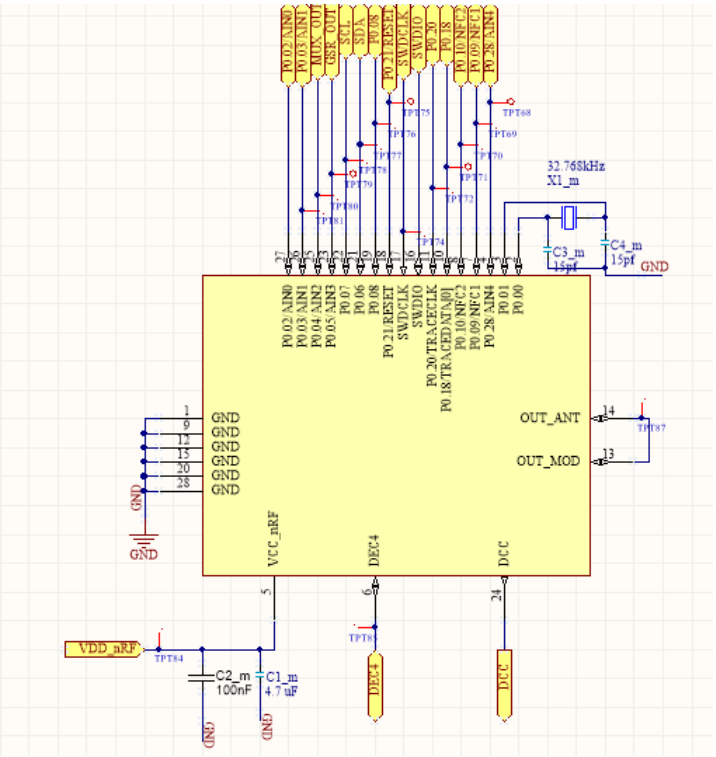


Figura 3.31: Schematico chip Taiyo

Memoria EEPROM La scheda prototipo forniva due soluzioni per la memorizzazione dei dati digitali raccolti dalle misurazioni eseguite con i diversi sensori : la prima rappresentata dall'uso di una memoria EEPROM collegata direttamente al BUS I2C (a sinistra del multiplexer nella gura 2.3); la seconda consiste nell'inserimento di una Micro USB esterna inseribile attraverso un connettore della Molex (in alro a destra, sopra il pulsimetro, nella gura 2.3), saldato sulla scheda e connesso al microcontrollore Aconno 52832 in modo seriale sfruttando il protocollo SPI.
In breve, la micro SD è una memoria ash di tipo NAND, di dimensioni 11mm x 15mm per 1 mm di spessore; presenta un pinout di 8 pins come mostrato nella seguente gura:

microSD Pinout, SPI Mode

Pin Number	Pin name	Signal Function
1	NC	No Connect
2	/CS	Chip Select
3	DI	Master Out/Slave In (MOSI)
4	Vdd	Supply Voltage 2.7v / 3.6v
5	CLK	Clock
6	Vss	Ground
7	DO	Master In/Slave Out (MISO)
8	RSV	Reserved

Figura 3.32: pinout Micro SD in SPI mode, [104]

La gura 3.32, indica appunto SPI mode, infatti il connettore micro SD è stato collegato, come slave (poichè la micro SD non può inviare dati verso il microcontrollore ma solo riceverli), al microcontrollore Aconno 52832 sfruttando le due linee di dati MOSI (Master Out Slave In) e MISO (Master In Slave Out) e la linea di CLK per la sincronizzazione dei dati stessi; il pin 6 è stato connesso a GND, che rappresenta appunta la quarta linea del protocollo SPI.
Lo schematico relativo al connettore SPI è indicato nella seguente gura:

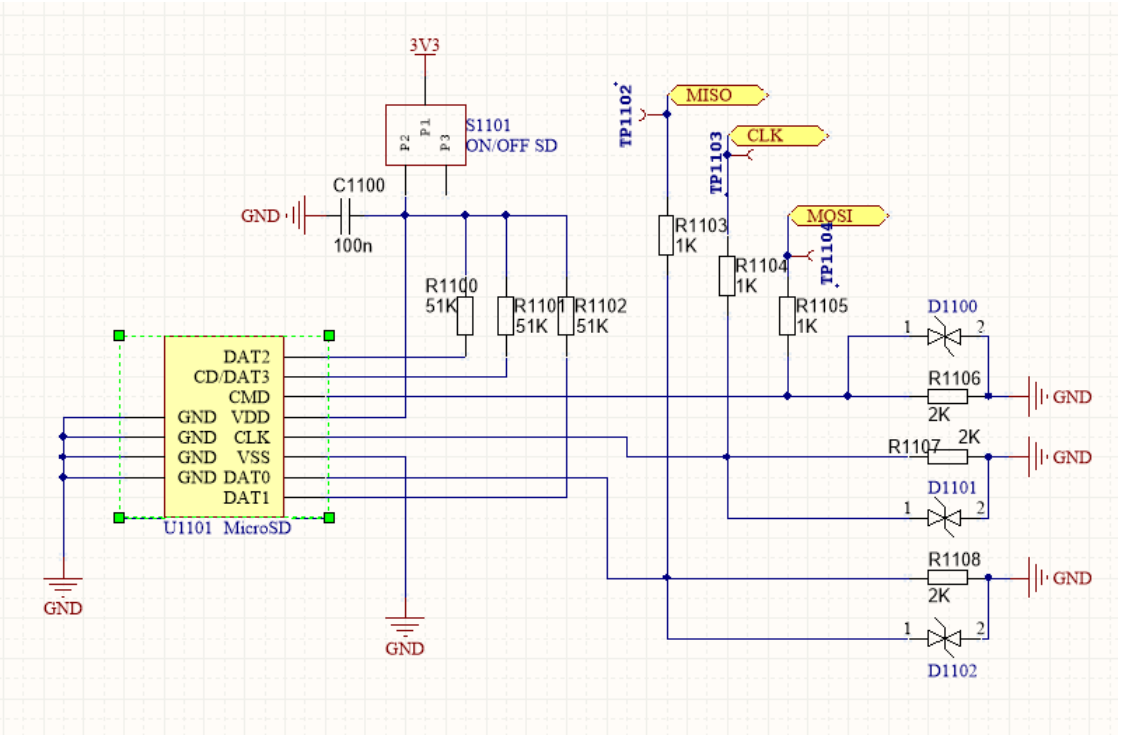


Figura 3.33: Micro SD schematico, [77]

Poichè, nella scheda miniaturizzata sono stati introdotti soltanto i componenti fondamentali e necessari al corretto funzionamento del Wired-glove, la micro SD è

stata considerata troppo ingombrante in termini di area occupata ed inoltre un elemento ridondante, in quanto già presente una memoria EEPROM come menzionato in precedenza. Dunque, nell'ottica di eliminazione delle ridondanze, la micro SD è stata eliminata , mantenendo soltanto la memoria EEPROM come dispositivo di memorizzazione dei dati campionati con i sensori. Dopodichè il microcontrollore leggerà i dati dalla memoria e li invierà via bluetooth, sfruttando l'antenna RF integrata, al pc dove verranno poi analizzati dal dottore o da colui che sta stabilendo il training riabilitativo al paziente.
La memoria EEPROM scelta, è la 24LC64 della Microchip technology. E' una memoria da 64Kbit organizzata come un singolo blocco da 8K righe con una word da 8 bits. Dunque, è possibile memorizzare sino a 8000 dati da 8bits, risultando particolarmente adatta per la nostra applicazione. Inoltre, i consumi di potenza sono bassi, in quanto la tensione di alimentazione varia tra 2.5 V e 5.5 V. Importante, connettere il pin WP (Write-protect) a GND per abilitare le operazioni di scrittura della EEPROM. In particolare, è stato scelto il package TSSOP a 8 pins:

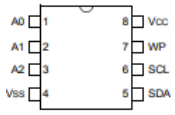


Figura 3.34: EEPROM package TSSOP, [105]

Lo schematico realizzato è indicato nella seguente gura :

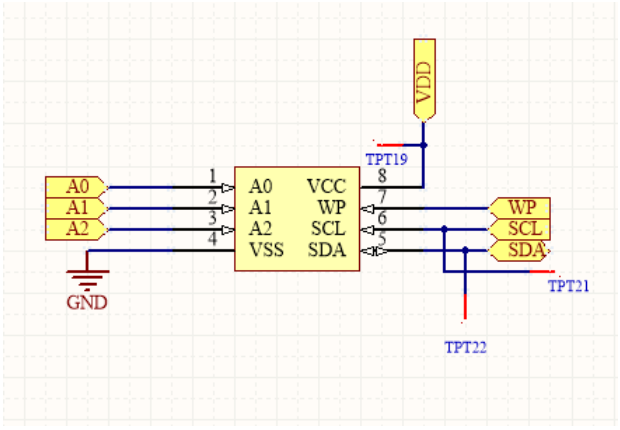


Figura 3.35: Schematico EEPROM

Un'altra modica, da evidenziare, rispetto alla scheda prototipale, è stata l'eliminazione di tutti gli switch (vedi gura seguente), che permettevano di abilitare/disabilitare manualmente quasi tutti i dispositivi presenti su tale scheda,

come : il magnetometro, l'accelerometro-giroscopio 3D, il pulsimetro, il sensore di temperatura, i sensori di bending e di forza e la memoria EEPROM stessa.

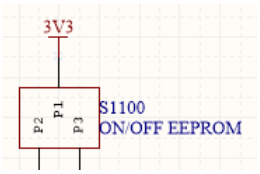


Figura 3.36: Switch della memoria EEPROM, [77]

Multiplexer Il multiplexer è un componente fondamentale per l'analisi dei segnali analogici provenienti rispettivamente dai quattro sensori di bending e dai quattro sensori di forza. Pertanto, è stato scelto un multiplexer con 8 ingressi e un'uscita, pilotato dal microcontrollore tramite tre bits S0, S1 ed S2. Il multiplexer scelto è il 74HC4051 prodotto dalla Nexperia.Questo dispositivo può essere usato sia per applicazioni analogiche, sia per applicazioni digitali; fa parte della tipologia degli switch SP8T (Single-Pole octal-through), dunque un multiplexer meccanico-digitale, dotato inoltre di un pin di enable $\overline{E}$ attivo basso, dunque quando è a '0' il multiplexer è attivo. Ammette un ampio range di segnali analogici d'ingresso con tensioni che possono variare tra -5V e +5V. Poichè il dispositivo è usato come multiplexer (da datasheet può essere usato anche come de-multiplexer), gli otto ingressi sono etichettati come Y0,Y1,...Y7 mentre l'uscita è etichettata come Z :

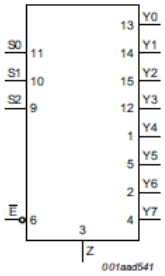


Figura 3.37: simbolo Multiplexer, [106]

Il package è di tipo QFN a 16 pins, di dimensioni 2.54x3.54 mm.
Lo schematico relativo al mux8-to-1 è riportato nella seguente gura:

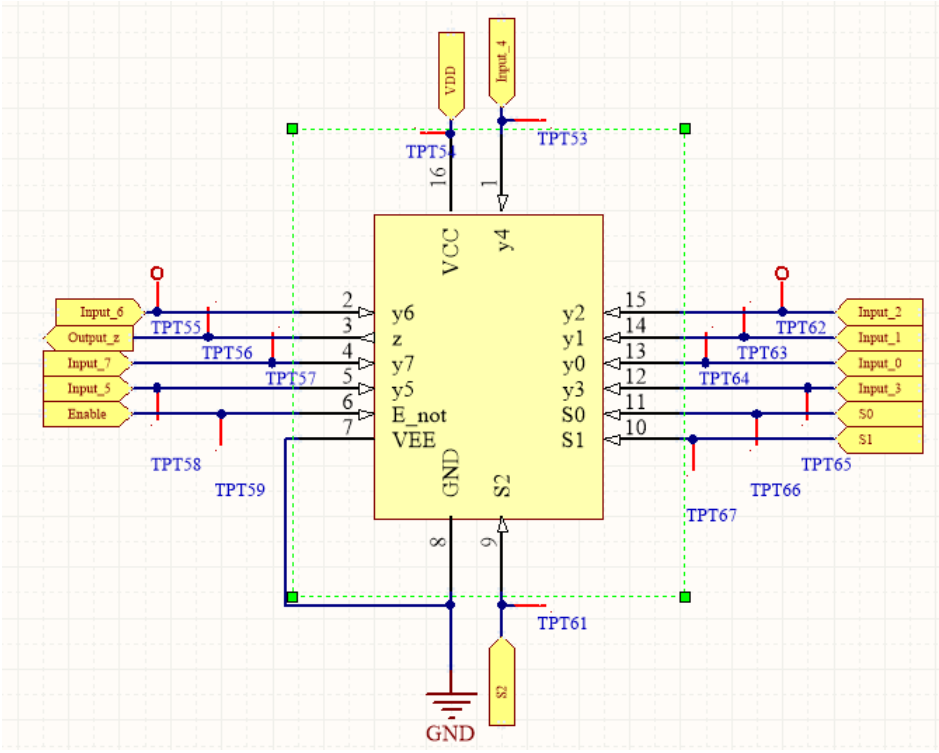


Figura 3.38: Schematico Multiplexer 8-to-1

3.2.2 Connessione componenti

Il secondo passo progettuale verso la denizione della PCB miniaturizzata nale, è rappresentato dalla realizzazione delle librerie di componenti passivi, che dovranno essere connessi ai componenti attivi.Le due librerie principali sono state chiamate Ceramic SMD capacitors e Resistors SMD, dove SMD è la sigla per Surface Mounted Device.Infatti, per mantenere ridotte le dimensioni della PCB, è stato necessario collocare i componenti sia su un Top Layer sia su un Bottom Layer e pertanto è stata adottata la tecnica del montaggio superciale, invece di quella basata sui fori passanti (through hole).
Dopodichè, sono stati realizzati gli outline drawings per gli altri elementi passivi che saranno collocati sulla board.Nella gura seguente è riportato un'esempio di PCB pattern di per un generico header, fondamentale per la denizione della footprint del componente:

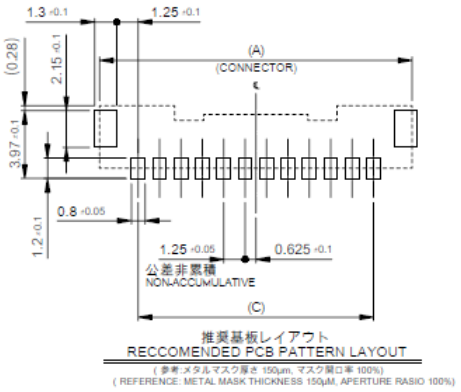


Figura 3.39: Header Molex, PCB pattern

Gli elementi passivi, diversi da condensatori e resistori sono: un header maschio verticale a 4 pins necessario per la programmazione della board; infatti, rappresenterà l'SWD (SerialWire Debug) header e dunque riceverà i segnali SWDIO, SWDCLK, VDD e GND dalla scheda con cui verrà programmata; un connettore SMD maschio a cui verrà collegata la batteria LiPo; un header maschio verticale a 2 pins a cui collegare il sensore GSR; un connettore femmina USB tipo B necessario per la connessione tra board e PC; un'oscillatore al quarzo da 32.768 KHz da connettere al microcontrollore Taiyo; un push-button per resettare il modulo bluetooth della Taiyo e due LEDs uno rosso e uno verde ciascuno pilotato, tramite un resistore da 560Ω dal pin d'uscita SYS del modulo TPS65721 di Power Management. I LEDs sono usati per indicare lo stato di carica della batteria, il rosso indicherà un livello di carica basso, mentre il verde rappresenterà una batteria quasi completamente caricaNelle seguenti gure sono mostrati i componenti appena citati:

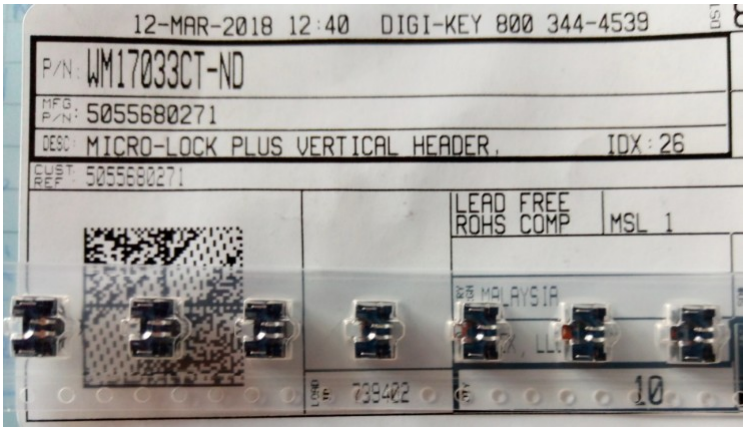


Figura 3.40: Header 2 pins verticale per il sensore GSR

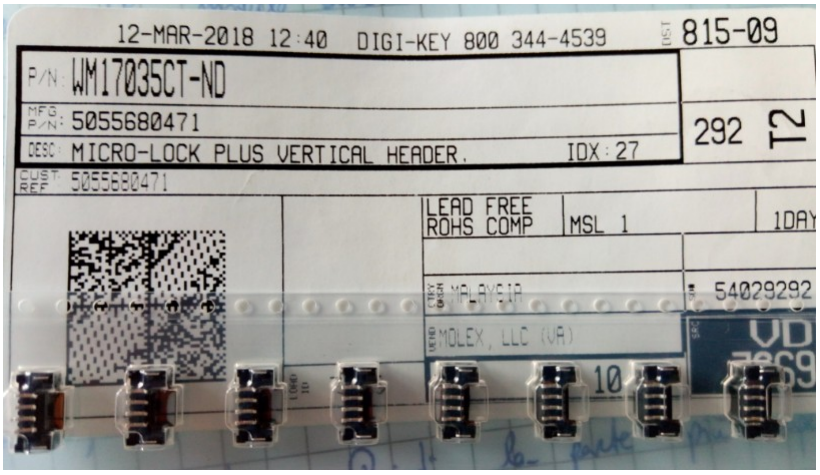


Figura 3.41: Header 4 pins verticale per la programmazione

Figura 3.42: Interruttore tattile di RESET

Figura 3.43: micro USB tipo B

Figura 3.44: Oscillatore al quarzo da 32.768 KHz

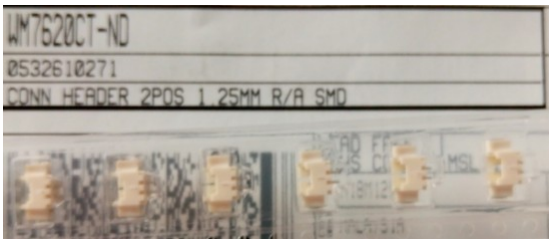


Figura 3.45: Connettore per batteria LiPo

Quindi, sono stati realizzati gli schematici di ciascun componente attivo, descritto nella sezione precedente. Da notare, la mancanza del sensore di temperatura e del pulsimetro; infatti, l'idea è stata di progettare una seconda board, più piccola e più semplice, su cui collocare questi due sensori. Tale scheda, verrà collocata in prossimità del polso del paziente in modo da poter misurare nella maniera più accurata possibile il battito cardiaco e in particolare i BPM impiegando il pulsimetro, e la temperatura corporea in prossimità dell'estremità superiore del paziente, su cui verrà collocato il wired-glove nale.
Lo step seguente, è stato il più complesso, ossia la denizione delle connessioni tra i diversi componenti. Iniziando, come in precedenza, dalla sezione di Power Management, il connettore micro USB tipo B e il connettore per la batteria LiPo sono stati connessi nel seguente modo ai pins del modulo TPS65721 :

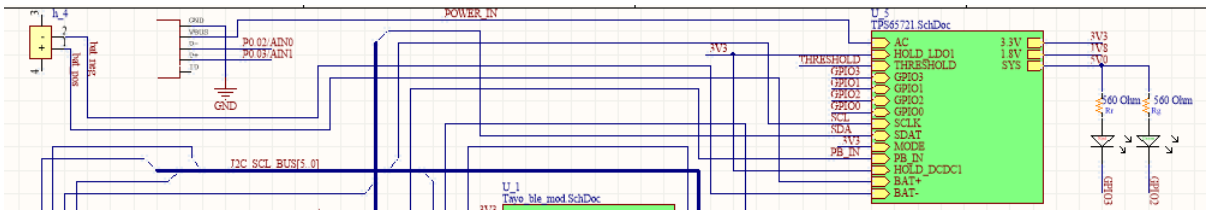


Figura 3.46: connessione sezione di Power Management

Il lead Vbus del connettore USB, che porta la tensione d'alimentazione di 5.0 V è connesso al pin AC d'ingresso del modulo TPS65721, mentre le due linee dati D- e D+ sono state collegate rispettivamente ai due pins di GPIO P0.02/AIN0 e P0.03/AIN1 del modulo bluetooth Taiyo. Invece, i due pins della batteria LiPo sono stati collegati ai pins BAT+ e BAT- del TPS65721. Importanti poi sono da evidenziare le tre uscite del blocco di gestione dell'alimentazione: 3.3V e 1.8V, la prima è usata per fornire la tensione di alimentazione alla board principale e alla board secondaria, mentre la seconda è usata per fornire la tensione d'alimentazione al pulsimetro collocato sulla board secondaria; SYS fornisce in uscita 5.0V ed è usata per pilotare i due LEDs di monitoring dello stato di carica della batteria.

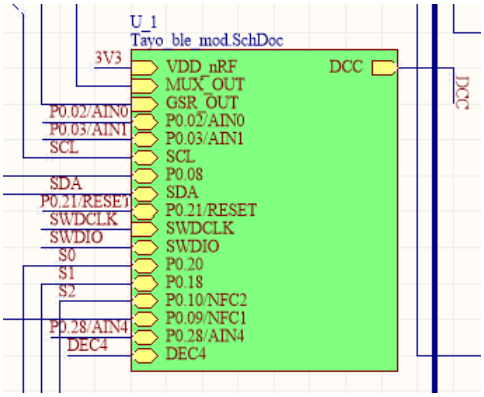


Figura 3.47: modulo Taiyo sheet symbol

Per quanto riguarda i componenti digitali, accelerometro-giroscopio 3D (LSM6DS0), magnetometro 3D (LIS3MDL) e memoria EEPROM 24LC64 sono stati collegati tutti ai due bus I2C SDA ed SCL, a cui sono stati collegati il microcontrollore stesso (vedi gura 3.47) e il modulo TPS65721 (vedi gura 3.46):

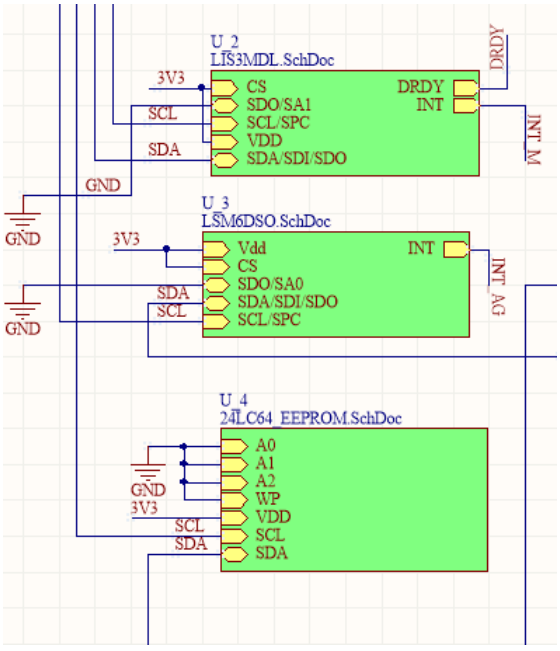


Figura 3.48: Sensori digitali, sheet symbols

Considerando, invece, i sensori analogici, le uscite dei quattro sensori di forza sono stati collegati ai primi quattro ingressi del multiplexer etichettati come Input0, Input1, Input2 e Input3 ; mentre le uscite dei quattro sensori di bending sono state collegate ai restanti quattro ingressi del multiplexer etichettati come Input4, Input5, Input6 e Input7. L'uscita del multiplexer *Outputz* è stata collegata in ingresso ad uno dei pin di GPIO del modulo ble Taiyo.I tre pins S0, S1 ed S2 di selezione del corretto ingresso del mux, sono pilotati rispettivamente dai tre pins del microcontrollore Taiyo, P0.20, P0.18 e P0.10/NFC2.Mentre l'ingresso di Enable del mux è pilotato dal pin P0.09/NFC1 del microcontrollore.

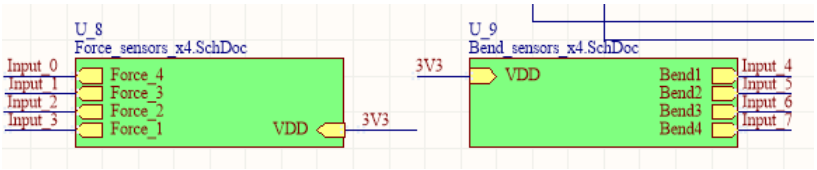


Figura 3.49: Sensori bending e forza, sheet symbols

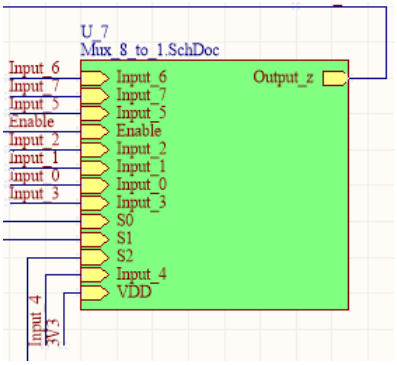


Figura 3.50: Mux8-to-1, sheet symbol

Inne, sono stati deniti i pads, collocati sul Bottom Layer della board, su cui verranno piazzati i pogo pins.Per pogo pins s'intendono particolari elementi meccanici a molla (spring-loaded).Si tratta di dispositivi metallici e dunque in grado di condurre segnali elettrici se compressi e messi a contatto con una supercie metallica esterna alla board. Nella seguente gura è mostrato il tipo di pogo-pin usato per questa PCB:

Figura 3.51: Pogo pin della Harwin, [107]

Dunque, poichè è stato realizzato un design di tipo gerarchico, è stato denito un Top levelschematic in cui sono indicate tutte le connessioni tra i componenti attivi dell'intera board:

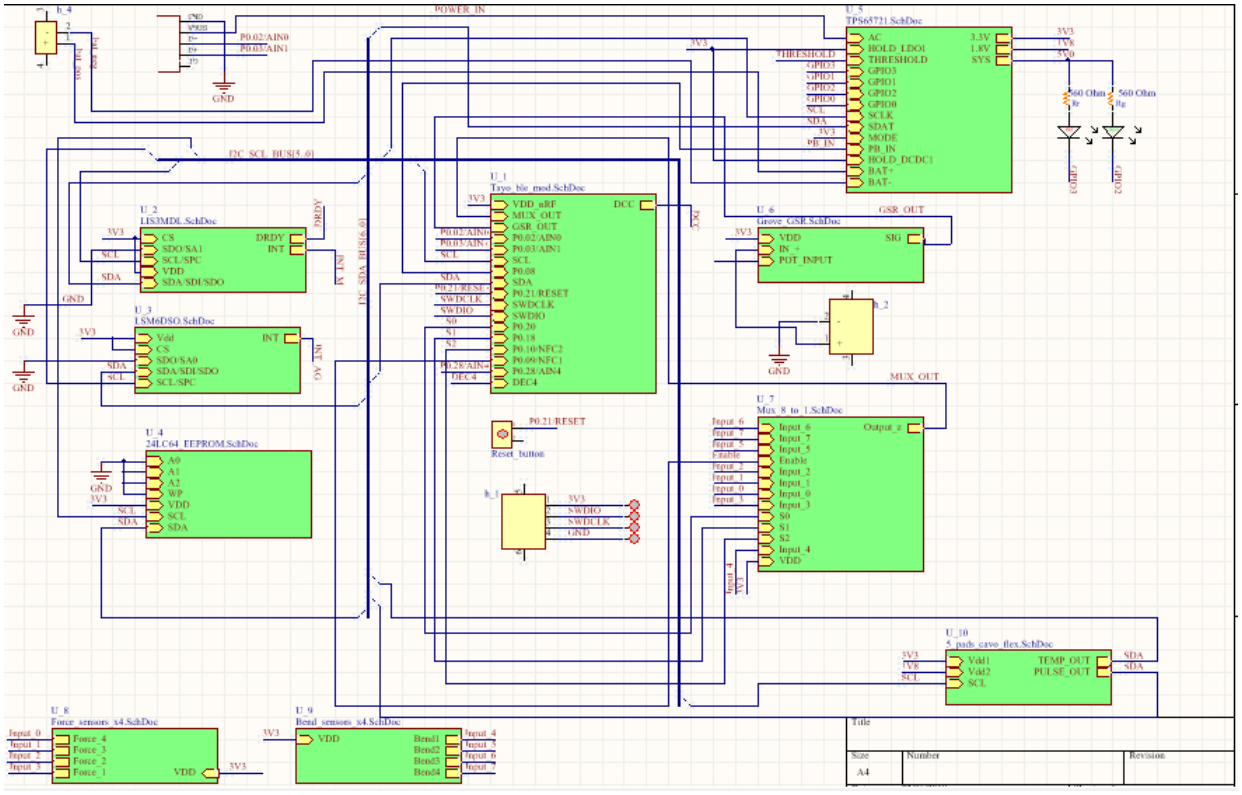
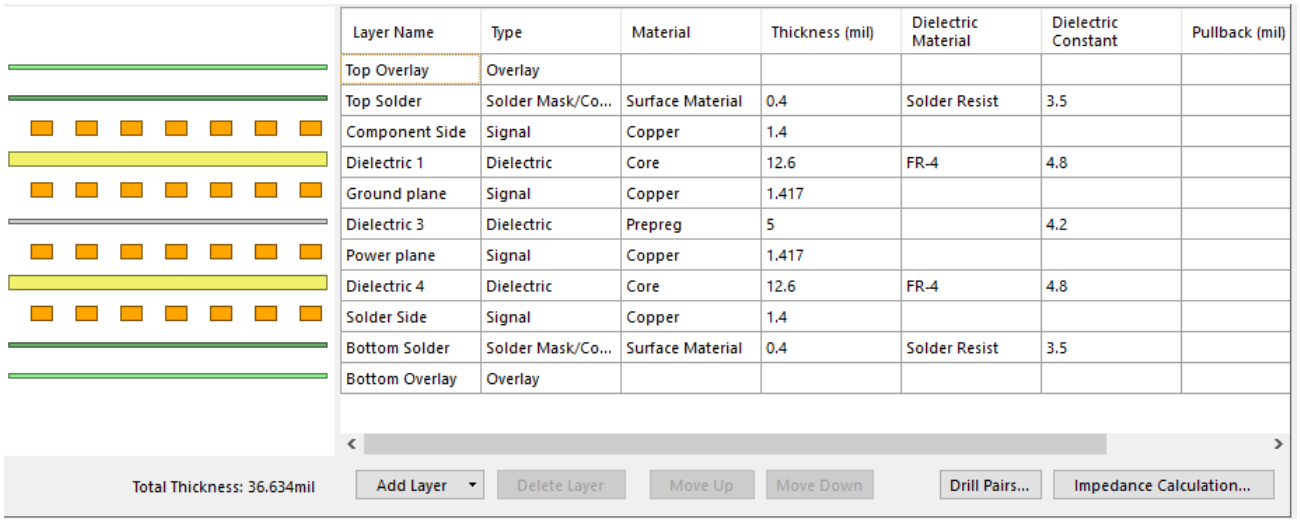


Figura 3.52: Top level schematic

A questo punto, deniti i componenti attivi e il modo in cui sono interconnessi fra loro, si è passati alla denizione del layout vero e proprio della PCB, che verrà montata sul wired-glove.Innanzitutto, è stata scelta la tipologia di PCB, ossia una board a 4 layers : un Top layer di segnale, un Bottom layer di segnale, un Power layer su cui far scorrere le linee della tensione di alimentazione, un Ground Layer.



Layer Name	Type	Material	Thickness (mil)	Dielectric Material	Dielectric Constant	Pullback (mil)
Top Overlay	Overlay					
Top Solder	Solder Mask/Co...	Surface Material	0.4	Solder Resist	3.5	
Component Side	Signal	Copper	1.4			
Dielectric 1	Dielectric	Core	12.6	FR-4	4.8	
Ground plane	Signal	Copper	1.417			
Dielectric 3	Dielectric	Prepreg	5		4.2	
Power plane	Signal	Copper	1.417			
Dielectric 4	Dielectric	Core	12.6	FR-4	4.8	
Solder Side	Signal	Copper	1.4			
Bottom Solder	Solder Mask/Co...	Surface Material	0.4	Solder Resist	3.5	
Bottom Overlay	Overlay					

Total Thickness: 36.634mil | Add Layer | Delete Layer | Move Up | Move Down | Drill Pairs... | Impedance Calculation...

Figura 3.53: Layer Stack Manager di Altium

La gura 3.53, mostra il Layer Stack Manager di Altium relativo alla board progettata. Si possono facilmente notare i quattro strati di segnali, appena descritti, ed indicati in gura con il type signal. Tra uno strato di segnale e il successivo, ci sono degli strati di materiale dielettrico; il più usato è la vetronite o vetro epossidico (berglass) di tipo FR-4 (dove FR sta per Fire Retardant, è infatti un materiale ignifugo). Infatti, la vetronite FR-4 è comunemente usata nelle PCB per costituire lo strato di CORE e gli strati di pre-preg. Per pre-pregs s'intende un materiale che è stato pre-impregnato con particolari resine sintetiche.
In questo caso, è stata denita una board con due strati di CORE, quindi a coppie di due sono stati posizionati i layers di segnale; in particolare, intorno al CORE Dielectric 1 sono stati collocati il Component Side e il Ground Plane, mentre attorno al CORE Dielectric 2 sono stati collocati gli strati di segnale relativi al Bottom layer, ossia il Power Plane e il Solder Side. La struttura a stack è poi completata da altre due coppie di strati, la Solder Mask indicata come Top Solder e Bottom Solder, ed inne lo strato di Overlay indicato come Top Overlay e Bottom Overlay.
Il Solder Mask è un rivestimento protettivo, usato su entrambi gli strati di segnale più esterni, per proteggere le tracce di rame dalla corrosione degli agenti atmosferici e da eventuali corto-circuiti che si potrebbero vericare durante l'operazione di saldatura dei componenti. Dunque, è uno strato che ricopre ogni parte del circuito, spazio vuoto e traccia di rame, ad eccezione dei pads scoperti su cui verranno saldati i componenti e le vias tra i diversi strati della PCB. Il solder mask più ecace è quello di colore verde. Invece, l'Overlay o strato di Silkscreen è usato per indicare il testo relativo al nome dei componenti; inoltre, è molto utile per denire i contorni dei land patterns dei componenti, in modo da facilitarne la saldatura soprattutto se fatta manualmente.
Lo spessore totale della PCB è indicato in basso a sinistra, nella gura 3.53, ed è pari a 36.364 mil. Per mil s'intende millesimo di pollice e per la conversione in unità metrica si sfrutta la seguente relazione:

$$1mil = 0.0254mm$$

di conseguenza lo spessore totale della board è all'incirca pari a 0.9 mm.
Dopodichè a livello di design, e dunque non si tratta di un layer sico vero e proprio, è stato introdotto il cosiddetto keep-out layer ossi un "contorno" usato per indicare il limite oltre il quale non può essere collocata alcuna traccia di rame e alcun componente.
Il passo di progetto successivo è stato il placing dei componenti, tenendo conto della posizione che la PCB avrebbe avuto sul dorso del wired-glove. Pertanto, considerato che i sensori analogici di bending e di forza dovranno essere collocati

sulle quattro dita della mano, ad eccezione del pollice, sulla parte superiore della board, sul Bottom side, sono stati collocati a coppie di 2, 16 pads per il piazzamento di altrettanti pogo-pins, 3.51.Infatti, ciascun sensore analogico è dotato di 2 pins, dunque 4 sensori di forza più 4 sensori di bending, si ha un totale di sedici uscite. Queste verranno collegate tramite altrettanti li al package in cui sarà alloggiata la board, andando poi a contatto con i pogo-pins e dunque trasferendo i segnali sulla PCB stessa, per poi essere analizzati. Invece, sul lato inferiore della board, sempre sul Bottom side sono stati collocati altri 5 pads per collocare altrettanti pogo-pins, sempre dello stesso tipo 3.51; in questo caso sono usati per trasferire cinque segnali dalla main board, mediante altrettanti cavi, verso la board secondaria su cui, come detto, sono stati collocati il sensore di temperatura MAX30205 e il pulsimetro MAX30105. Dopodichè, sul lato sinistro della board, sul Component Side, sono stati collocati in alto l'header a 4 pins per la programmazione della board e in basso il connettore micro-usb tipo b. Sul lato destro, invece, sempre sul Component Side, sono stati collocati: in alto l'header a 2 pins per il collegamento del sensore GSR e in basso il connettore per la batteria LiPo. Sempre sullo strato Component Side sono stati collocati la maggiorparte dei componenti attivi ossia: il TPS65721, il multiplexer 8-to-1, il magnetometro LIS3MDL, l'MCP6004 che realizza il circuito di condizionamento per il sensore GSR, l'accelerometro-giroscopio 3D LSM6DS0, la memoria EEPROM e il Microcontrollore bluetooth della Taiyo.
Il componente dotato della footprint più complessa da realizzare è stato sicuramente il Microcontrollore. Infatti, è caratterizzato da un'elevato numero di pins, non tutti equispaziati fra loro, di cui la maggiorparte presenta una forma rettangolare, tuttavia alcuni hanno forma circolare. Una parte, poi è quasi completamente priva di pins, poichè è la regione in cui si trova l'antenna RF integrata. La denizione della footprint si è basata sul seguente mechanical outline drawing:

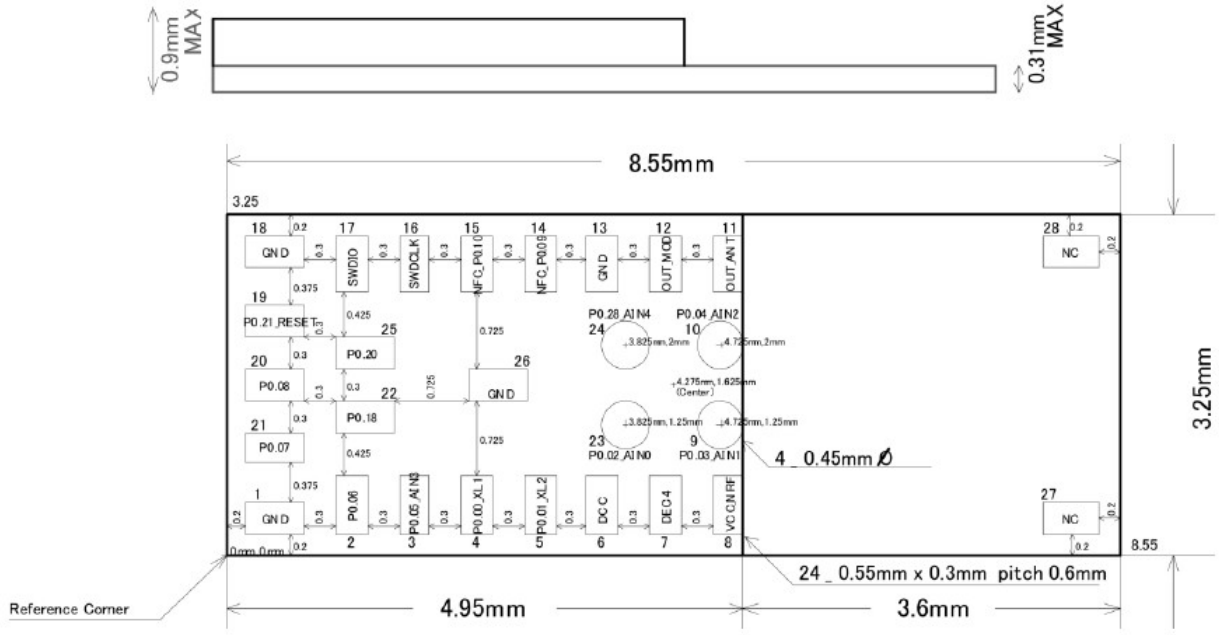


Figura 3.54: Outline meccanico della footprint del microcontrollore, [103]

Mentre la corrispondente footprint denita con Altium è mostrata nella seguente gura:

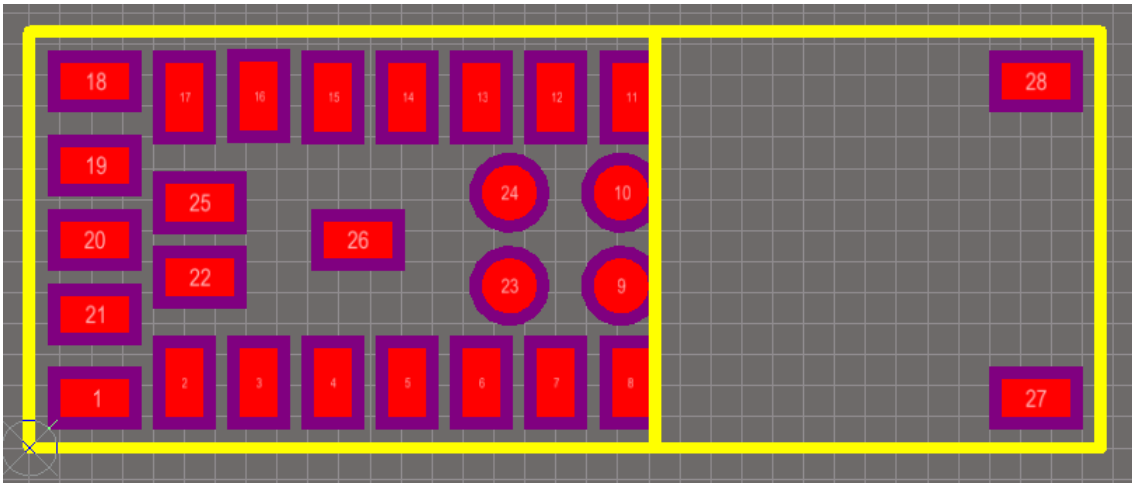


Figura 3.55: Microcontrollore Taiyo footprint

Dopodichè, è stato necessario seguire alcuni vincoli riguardanti il piazzamento del Micro sulla PCB, come indicato sul datasheet [103], soprattutto per garantire il corretto funzionamento dell'antenna RF per la comunicazione dati attraverso il protocollo Bluetooth. La seguente gura mostra i tre principali vincoli :

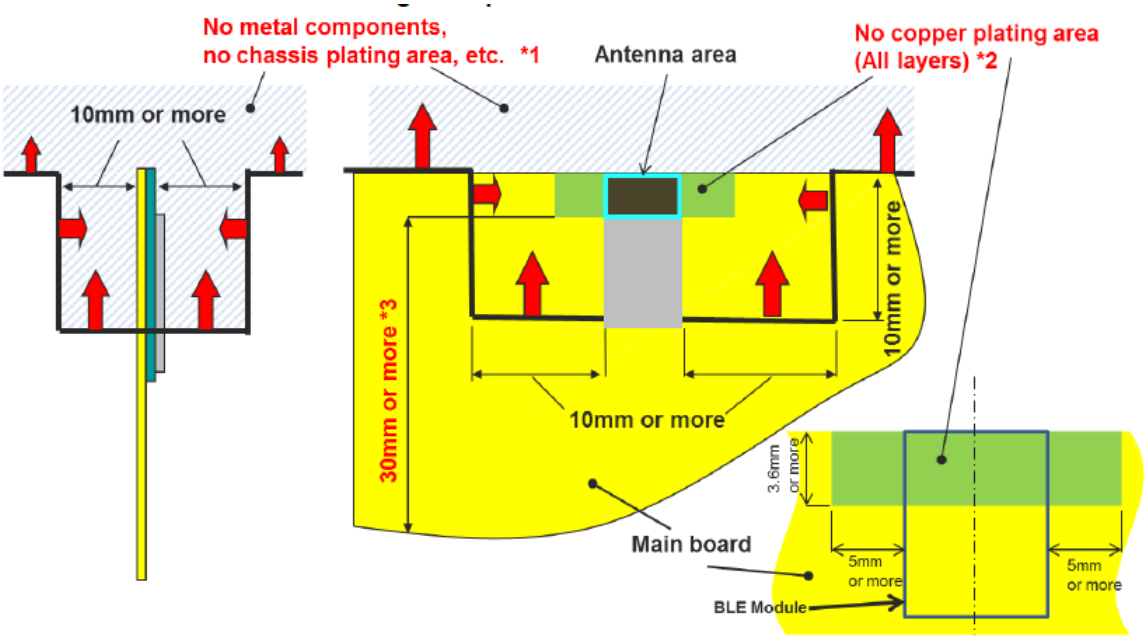


Figura 3.56: regole piazzamento microcontrollore, [103]

La regola *1 consiste nel non piazzare alcun componente metallico nella zona ombreggiata di blu, ad eccezione ovviamente dei componenti che dovranno essere saldati sulla PCB stessa.Poi, la regola *2 stabilisce che nel rettangolo verde, di dimensioni indicate in gura 3.56, non devono essere collocate nè tracce di rame, nè layer di rame. Può esserci una zona di materiale dielettrico FR-4, in quanto l'antenna bluetooth funziona con tale tipo di materiale. L'ultima regola, la *3, consiste nella denizione di un piano di GND che abbia una lunghezza minima di 30 mm a partire dal vertice basso del rettangolo verde 3.56, per permettere un buon funzionamento dell'antenna RF. Pertanto, è stato deciso di collocare il microcontrollore nella parte superiore della board in modo da avere un GND pattern il più lungo possibile. Quindi è stato realizzato un cut-out, o meglio uno per ciascuno dei quattro layer della board di dimensioni pari a quelle del rettangolo verde indicato in gura 3.56, in modo da non avere nessun strato metallico in quella zona.
Dopodichè, in basso a destra è stato collocato il modulo TPS65721 di Power Management; alla sua sinistra è stato messo l'accelerometro-giroscopio 3D LSM6DS0. Al centro sono stati posizionati il multiplexer 74HC4051 e il magnetometro LIS3MDL; mentre in alto a destra sono stati disposti la memoria EEPROM 24LC64 e il quadruple amplier MCP6004 relativo al sensore GSR della Seed StudioLa seguente gura mostra il layout del Component Side della PCB, in cui sono presenti i componenti appena descritti:

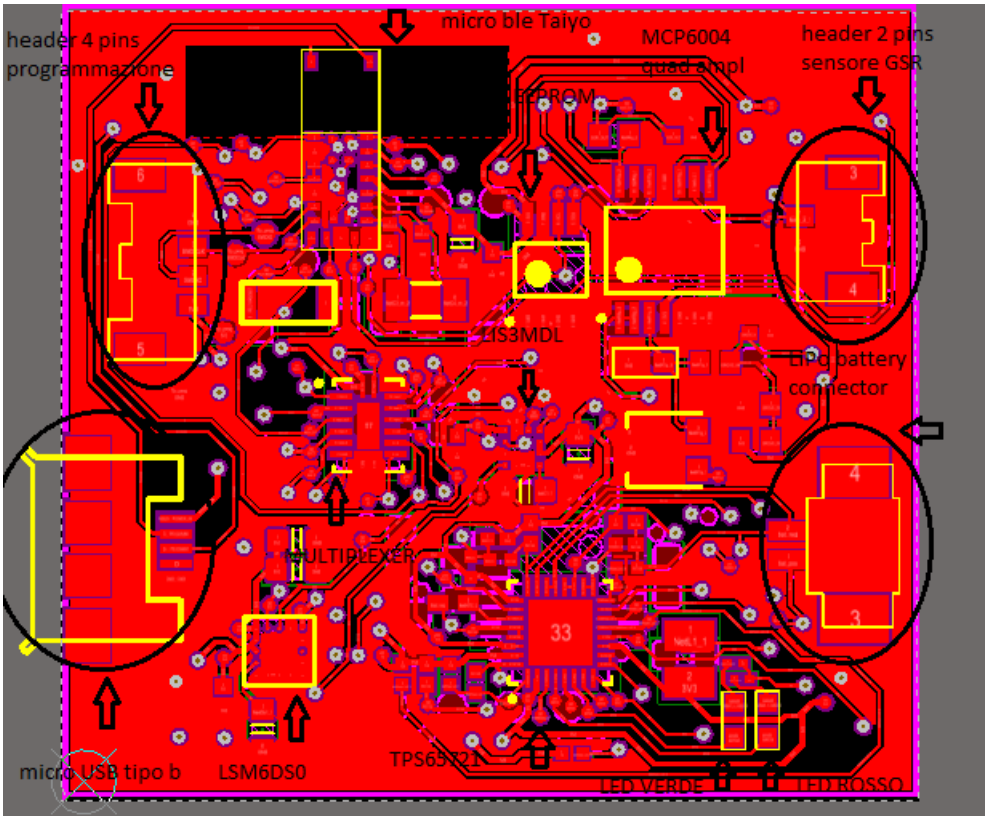


Figura 3.57: main PCB, Component Side

Per quanto riguarda il Bottom Layer, invece, come detto sono stati inseriti i pogo pins, [107] per ricevere i segnali provenienti dai sensori di forza e di bending. Di conseguenza, su questo lato della board sono stati collocati due MCP6004 (amplicatori quadrupli) per denire i circuiti di condizionamento rispettivamente per i quattro sensori di forza e per i quattro sensori di bending. Al di sopra della schiera dei 16 pads per i pogo-pins si trova il cut-o relativo al layer di Solder Side. In basso, invece, ci sono i 5 pads, per il collocamento di altrettanti pogo-pins, per l'invio dei segnali verso la board secondaria. La gura seguente mostra il layout del Solder Side della PCB:

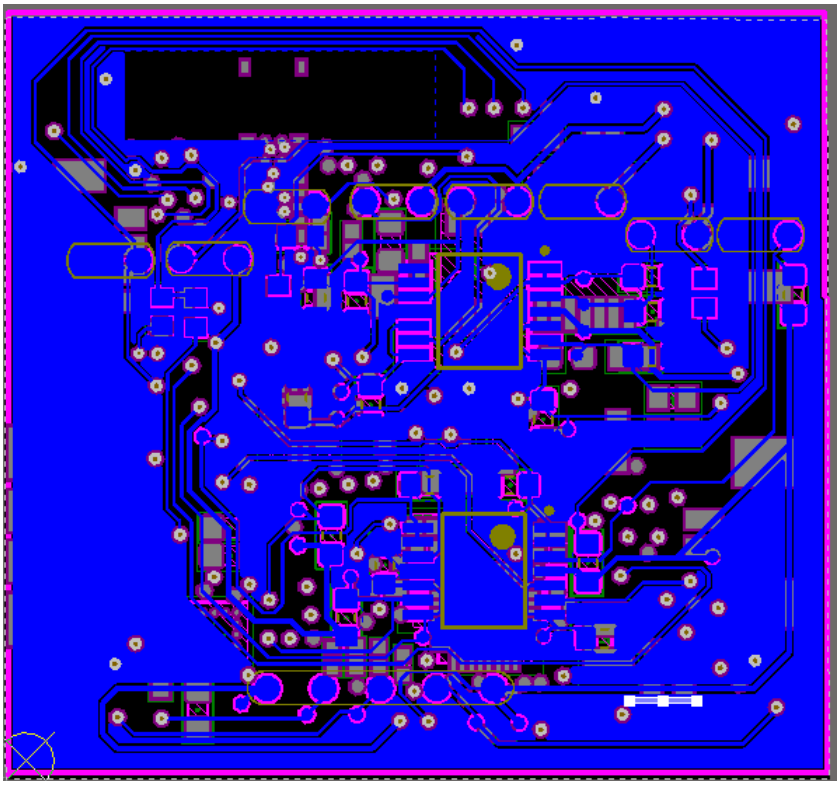


Figura 3.58: main PCB, Solder Side

Da sottolineare, che la linea di colore viola tutt'intorno al prolo della PCB, che si può notare sia nell'immagine 3.57, sia nell'immagine 3.58, rappresenta il keep-out layer di cui si è parlato in precedenza.
Lo step di progetto seguente, è stato la connessione dei diversi componenti tra loro, rispettando quanto stabilito nel Top levelschematic, vedi 3.52. Nonostante, un'attento piazzamento dei componenti, lo step d'interconnessione è stato il più dicile ed ha richiesto una maggiore quantità di tempo per poter essere portato a termine. Ciò è dovuto, soprattutto, alle dimensioni ridotte della board e dunque al poco spazio disponibile lungo cui collocare le tracce di rame. In particolare, è stato tenuto conto delle design rules imposte dal produttore di PCB, dove a design terminato sarebbe stata prodotta la PCB. Il produttore scelto, soprattutto per il ridotto costo di produzione (infatti, i tempi di produzione e consegna sono stati piuttosto lunghi) è stato Seed Studio. Perciò è stato necessario rispettare tali regole per garantire la realizzazione della board. Le rules principali sono state riportate nelle seguenti immagini:

Items	Description	Specs
		Unit: mm
Board Dimension	Min size	10mm*10mm Tip: If your board width is smaller than this size, you can make a bigger panel and use slots to separate the board.
	Max size	500*500mm
Available Board Layers	1-6 layers	
Available Board Qty	Min：5pcs	
	Max: 8000pcs	
Dielectric Constant	4.2-4.7	
Dielectric Separation thickness		0.075 - 5.0
Available Board Thickness	1-2 layers	0.6, 0.8, 1, 1.2, 1.6, 2, 2.5, 3
	4 layers	0.8, 1, 1.2, 1.6, 2, 2.5, 3
	6-8 layers	1, 1.2, 1.6, 2, 2.5, 3
	10 layers	1.2, 1.6, 2, 2.5, 3
	12 layers	1.6, 2, 2.5, 3
	14 layers	2, 2.5, 3
	16 layers	2.5,3
Available Board Copper Weigh	1oz. 2oz. 3oz.	

Figura 3.59: Seed Studio rules, parte 1 , [108]

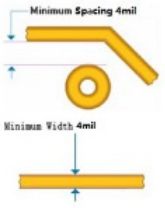
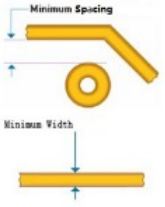
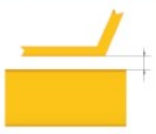
Board Thickness Tolerance	± 10%	
Minimum trace spacing / width		For 1oz, 4/4mil, 5/5mil, 6/6mil For 2oz, 10/10mil For 3oz, 15/15mil
Minimum trace width in inner layers (for 4 layers board)		≥6mil
Minimum distance between trace and copper pour		For 1oz ≥ 8mil For 2oz ≥ 12mil For 3oz ≥ 15mi

Figura 3.60: Seed Studio rules, parte 2, [108]

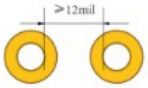
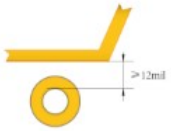
Minimum distance between trace and copper pour		For 1oz ≥ 8mil For 2oz ≥ 12mil For 3oz ≥ 15mi
Minimum distance between vias (plated holes)		≥12mil Aim to prevent Ion migration
Minimum distance between via(plated holes) and trace		≥12mil Aim to prevent Ion migration
Annular Rings		≥0.152mm/6mil

Figura 3.61: Seed Studio rules, parte 3, [108]

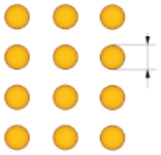
Outer Layer Copper Thickness		0.035-0.07(1oz-2oz)
Inner Layer Copper Thickness		0.017(0.5oz)
Drilling Hole Diameter(Mechanical)		0.2 - 6.3mm
Width of Solder Mask Dam		Normal: ≥0.32mm for Green ≥0.35mm for Other colors Limitation(need extra fee) ≥0.10mm for Green ≥0.13mm for Other colors
Diameter of Castellated Holes		≥0.50mm
Size of BGA		For 6/6mil ≥0.45mm For 5/5mil ≥0.35mm For 4/4mil ≥0.25mm

Figura 3.62: Seed Studio rules, parte 4, [108]

Innanzitutto, occorre specicare che per le PCB, lo spessore del rame è espresso in

once (oz). S'intende, appunto, lo spessore ottenuto quando 1 oz di rame è pressata ottenendo una supercie piatta e distribuita in modo uniforme su un'area di un piede quadrato (square foot)Tale spessore, risulta pari a 1.37 mils (millesimi di pollice); convertendo dalle unità imperialial sistema metrico decimale, 1 oz di spessore corrisponde a circa 0.035 mm.
Quindi, per quanto riguarda la realizzazione delle tracce di rame e delle vias, sono state considerate le tre seguenti rules, denite nella gura 3.60 e nella gura 3.61: Minimum trace spacing/width di 4mil/4mil per rame da 1 oz ; Minimum trace width in inner layers e Annular rings. Dunque, nelle Design/Rules di Altium sono stati settati i seguenti parametri: per le Vias rispettivamente un Diameter (D) pari a 23.622 mil e un Hole Size (HS) pari a 9.842 mil in modo da ottenere un Annular ring pari a $\frac{D-HS}{2} = 6.89 mil$; una track width per le linee di segnale pari a 4mil; una track width per le linee di alimentazione pari a 10mil, considerato che su tali tracce scorrono correnti di valore più elevato.Da sottolineare, inoltre, che sono stati rispettati gli spessori dei layers di rame Interni (Inner) ed Esterni (Outer), come mostrato nella seguente Layer Stack Table:

Layer	Name	Material	Thickness	Constant
1	Top Overlay			
2	Top Solder	Solder Resist	0,010mm	3,5
3	Component Side	Copper	0,036mm	
4	Dielectric 1	FR-4	0,320mm	4,8
5	Ground plane	Copper	0,036mm	
6	Dielectric 3		0,127mm	4,2
7	Power plane	Copper	0,036mm	
8	Dielectric 4	FR-4	0,320mm	4,8
9	Solder Side	Copper	0,036mm	
10	Bottom Solder	Solder Resist	0,010mm	3,5
11	Bottom Overlay			

Figura 3.63: Altium layer stack table

La gura 3.63 mostra, infatti, come gli strati più esterni, Component Side e Solder Side, abbiano uno spessore di 0.036 mm; allo stesso modo, gli strati più interni, Power plane e Ground Plane, presentano uno spessore di 0.036 mm. Tali valori soddisfano le Rules indicate in gura 3.62.
Le dimensioni nali della board sono di 36.95x33.9 mm, dunque una supercie piuttosto piccola. Ciò ha portato alla realizzazione di un gran numero di Vias, passanti per tutti e 4 i layers, non solo per portare i segnali sul Solder Side, ma anche per denire alcune tracce di segnale sui due strati più interni di Power Plane e Ground Plane. Il Power Plane è stato suddiviso in 3 regioni distinte, tramite

altrettanti Poligon pours, per poter supportare tre diverse tensioni di alimentazione:3.3 V, 1.8 V e 5.0 V. Pertanto, su tale strato sono state denite le tracce di rame corrispondenti a questi valori di tensione, oltre appunto ad alcune tracce di segnale che non sono state collocate sul Component Side layer per mancanza di spazio.

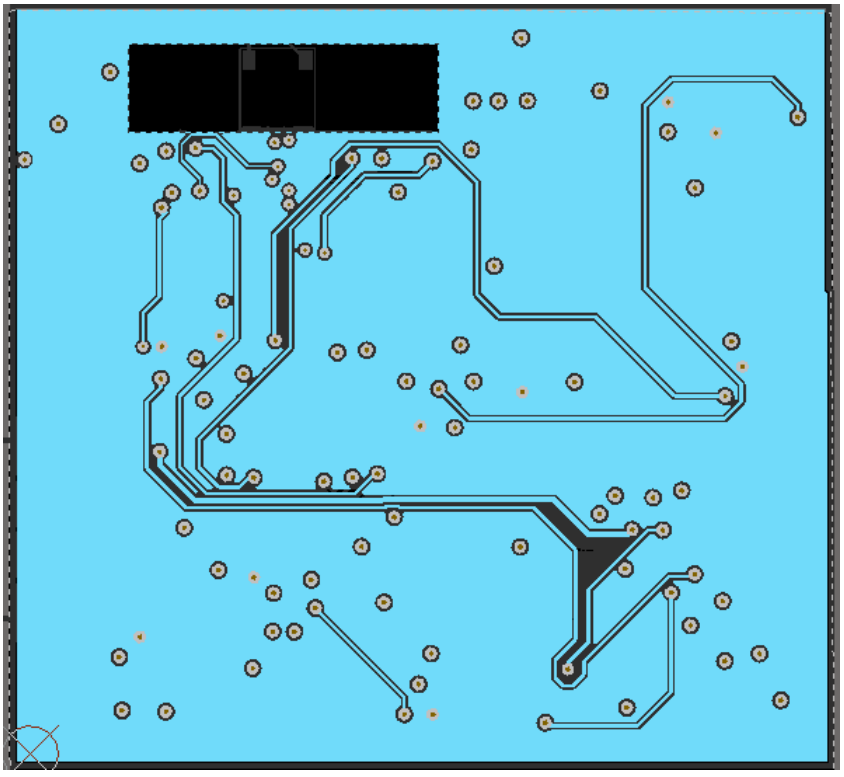


Figura 3.64: main PCB,Power Plane

Inne, è stato denito uno strato di GND, comune per tutti i layers, a cui sono stati collegati tramite Vias i piani di GND degli altri strati della board. Tuttavia, anche su questo layer, che di norma è lasciato privo di alcuna traccia metallica, sono state introdotte alcune tracce di segnale, sempre per gli spazi ridotti.

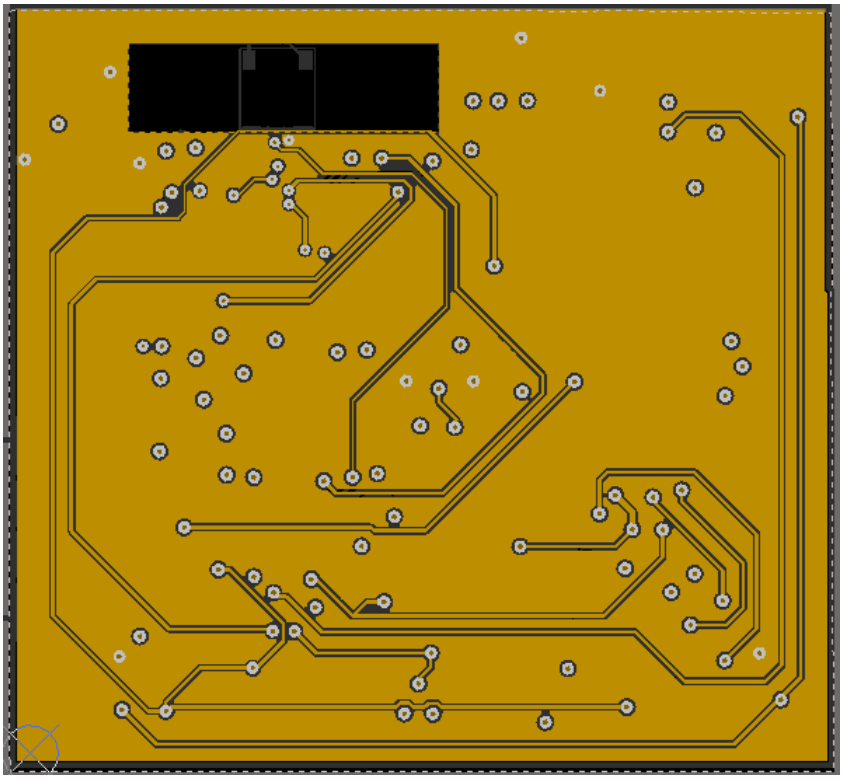


Figura 3.65: main PCB, Ground Plane

Una volta terminato il Routing di tutti i componenti, sono stati collegati inne i pins di GND realizzando un piano comune noto come Poligon pour di GND. Ciò è stato realizzato per entrambi i layer di segnale Component Side e Solder Side, denendo un poligon pour per ciascuno strato.

Capitolo 4

Capitolo 4: Realizzazione della PCB miniaturizzata

Dopo aver completato il design della PCB, questo è stato inviato presso il manufacturer cinese Seed Studio per la realizzazione della PCB. E' stato scelto tale manufacturer per i suoi costi ridotti rispetto ai suoi principali concorrenti, come ad esempio Eurocircuits.Tuttavia, i tempi di produzione e consegna sono stati piuttosto lunghi. Infatti, dopo 1 mese la PCB ci è stata consegnata, ed è mostrata nella seguente gura:

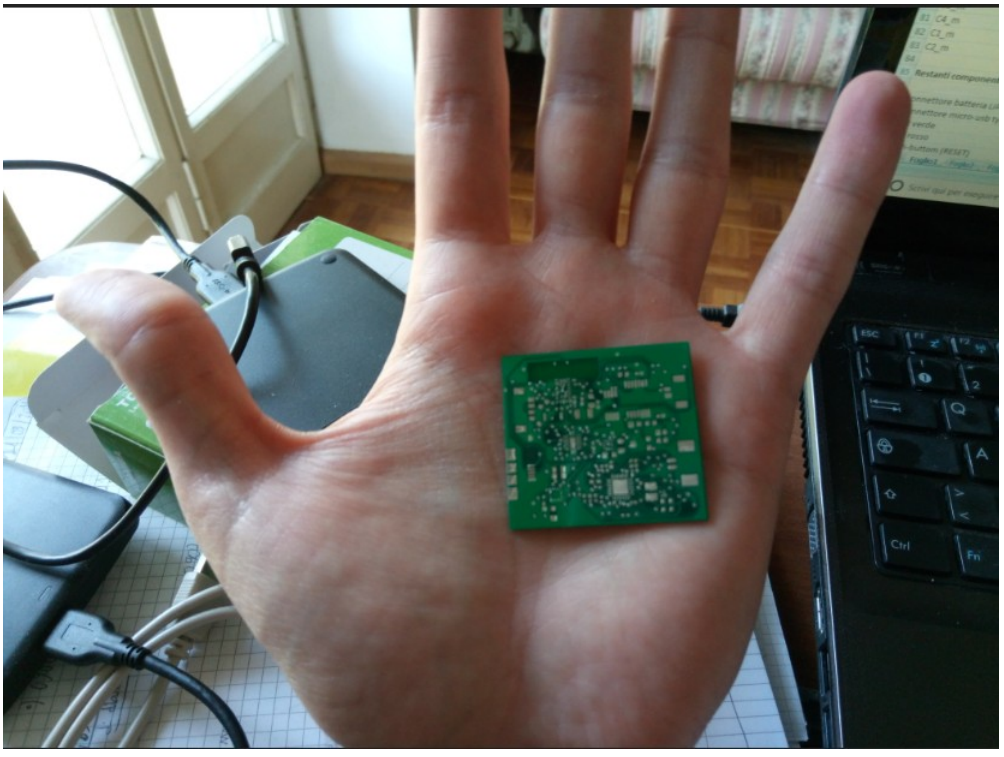


Figura 4.1: main PCB, prodotta da Seed Studio

La gura 4.1, mostra chiaramente le dimensioni ridotte della board, semplicemente comparandola con la mia mano in cui è mantenutaLo step successivo, è rappresentato dalla saldatura manuale di tutti i componenti attivi con i relativi componenti passivi, descritti nel Capitolo 3, in modo da ottenere la versione completa della board progettata.Gli strumenti usati per il completamento di tale operazione, sono i seguenti:

- Unità di alimentazione Weller WD 2M, a due uscite [119];
- Kit di saldatura Weller WP80 per stazioni di saldatura con sistema WDH IOT safety rest, [117];
- Saldatore di tipo Pencil WP80 con due punte dierenti: a scalpello dritto, a punta conica dritta, [116];
- Pinzette per dissaldatura Weller T0051317899N, tensione di alimentazione 12 V, potenza 40 W, [120];
- Presselle;
- Bobina di stagno;
- Pasta saldante Mechanic;
- Bomboletta spray rimuovi ussante Electrolube HFFR400DB ;
- Penna ussante;
- Flussante "gel" in siringa da 10g;
- Sostegno per riparazioni PCB, [118];
- Pistola saldatura ad aria calda;
- videocamera USB + eseguibile Oasis;

Nelle seguenti foto sono riportati parte degli strumenti usati per lo step di saldatura:

Figura 4.2: stazione saldante Weller

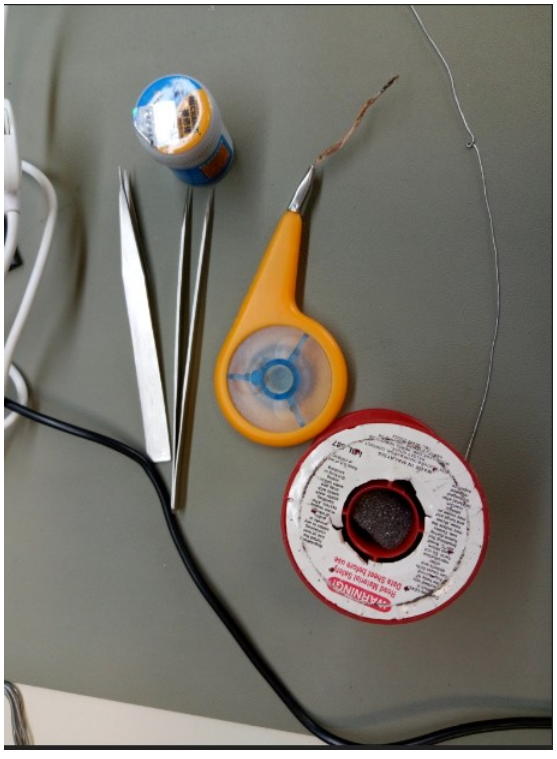
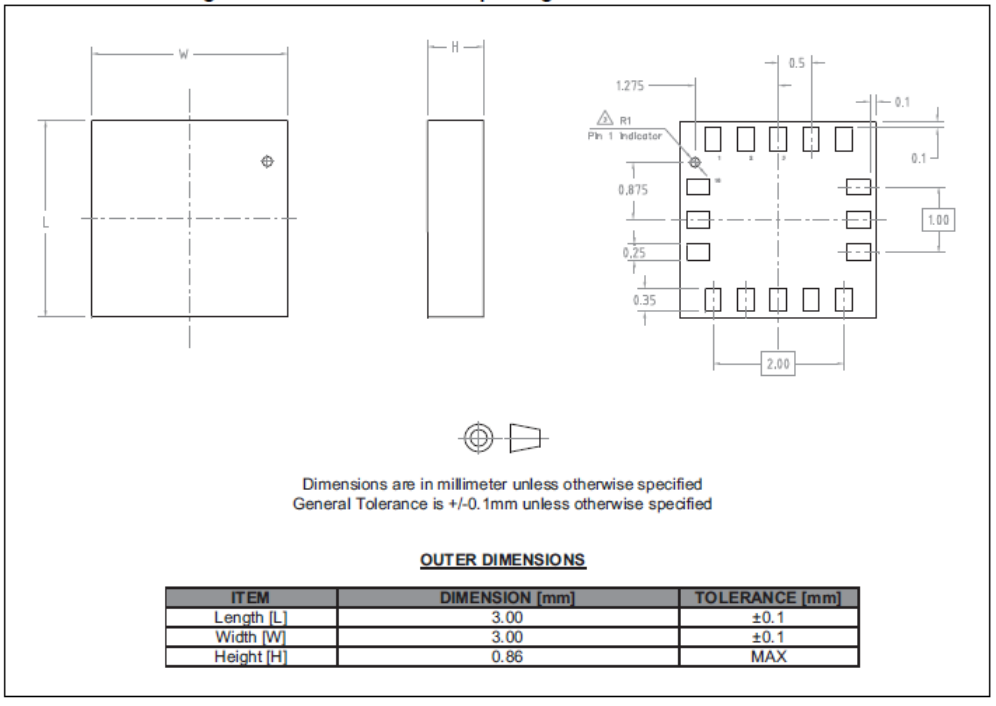


Figura 4.3: Strumenti saldatura

Per quanto riguarda la saldatura dei componenti passivi, la procedura adottata è stata pressochè la stessa per tutti. Da evidenziare, che per poter eseguire la saldatura di componenti così piccoli, è stato necessario sfruttare una videocamera collegata via USB al pc, quindi sfruttando il programma Oasis è stato possibile garantire lo zoom opportuno per rendere i componenti maggiormente visibili. I passi seguiti sono stati: applicazione del ussante sulle pads metalliche relative al componente da saldare, posizionamento del componente sulle pads corrispondenti, impiegando una pressella, usare una goccia di stagno sulla punta del saldatore e procedere inne con la saldatura. Per ripulire la zona circostante la saldatura dell'alone lasciato dal ussante e dell'eventuale sporco presente sulla PCB, s'impiega la bomboletta spray rimuovi ussante. Invece, per la saldatura dei componenti attivi è stato un po' più complicato; soprattutto poichè non tutti i componenti presentavano una footprint di tipo QFN, WQFN o simile e dunque con i leads che sporgono al di fuori del componente stesso. I chips, più dicili da saldare sono stati, infatti : l'LSM6DS0 dotato di footprint LGA a 16 pins, il LIS3MDL dotato di footprint LGA a 12 pins e il Microcontrollore Taiyo. Queste tipologie di footprint, infatti, presentano i pins al di sotto del chip stesso, dunque sia la fase di posizionamento dell'integrato sulle pads metalliche sia la fase di saldatura risutano più complicate. Come esempio di package di tipo LGA è stato riportato il package outline dell'accelerometro-giroscopio 3D:

Figure 24. LGA 3x3x0.86 16L package outline and dimensions



OUTER DIMENSIONS

ITEM	DIMENSION [mm]	TOLERANCE [mm]
Length [L]	3.00	±0.1
Width [W]	3.00	±0.1
Height [H]	0.86	MAX

Figura 4.4: LSM6DS0 package outline, [79]

Pertanto la strategia adottata per la saldatura di tali componenti è stata: applicazione della penna ussante, dopodichè al centro della footprint dell'integrato è stato applicato il ussante "gel" tramite la siringa, ciò è fatto per favorire una migliore adesione del chip sulla PCB. Dopodichè, è stata disposta la pasta saldante su ciascuna pad metallica della footprint del componente da saldare; quindi aiutandosi con la pressella il componente è stato correttamente posizionato. Quindi, mantenuto in tale posizione, sempre con l'aiuto della pressella, è stata usata la pistola di saldatura ad aria calda per andara a scaldare la pasta saldante sino al punto in cui dallo stato solido diventa stagno liquido e salda di conseguenza il componente.
L'immagine seguente mostra il Microcontrollore Taiyo, 3.27, appena saldato:

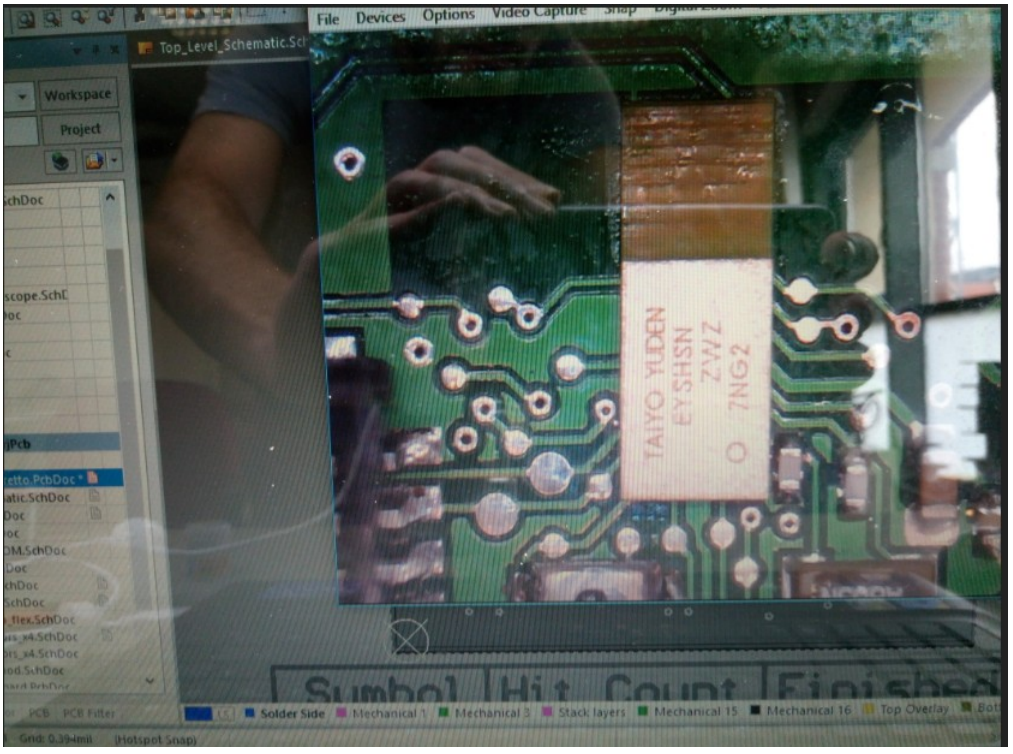


Figura 4.5: Microcontrollore visto sullo schermo del Pc, grazie alla videocamera USB collegata

Dopo aver terminato la saldatura dei componenti sul Component Side, si è passati al Solder Side. Questa seconda fase è stata più breve, per il minor numero di componenti passivi e soltanto due attivi, i due quadruple amplier MCP6004. Tuttavia, anche il Solder Side prevedeva dei componenti dicili da saldare, ossia i pogo-pins [107]. Infatti, si tratta di elementi molto piccoli e dalla forma irregolare che non ha facilitato nè la fase di posizionamento nè la fase di saldatura. Con pazienza, è stata applicata la pasta saldante su ciascuna pad circolare dei

pogo-pins (da ricordare, sono 16 sul lato superiore e 5 sul lato inferiore della PCB). Quindi, è stato eseguito il posizionamento, dopodichè sfruttando la pistola di saldatura ad aria calda sono stati saldati alla board.
Terminata la saldatura, la PCB nale ottenuta è stata riportata nelle seguenti gure:

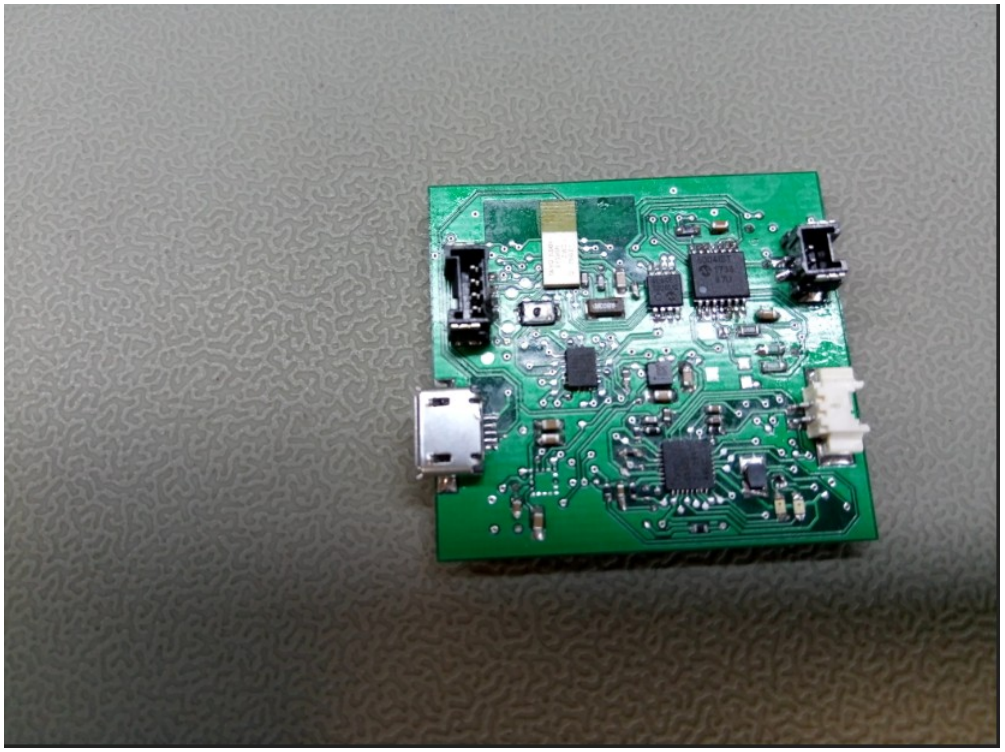


Figura 4.6: PCB nale, Component Side

Figura 4.7: PCB nale, Solder Side

4.1 Conclusioni

In conclusione, l'obiettivo nale di questo progetto consisterà nella realizzazione della seconda PCB su cui sono stati collocati il sensore di temperatura MAX30205 e il pulsimetro MAX30105 e collegarla alla principale qui descritta.La PCB secondaria sarà collocata in prossimità del polso del paziente per una corretta misurazione del battito cardiaco (e di conseguenza dell'heart rate). E' importante da sottolineare che la PCB principale, verrà inserita all'interno di un package, la cui forma è stata decisa basandosi sul dispositivo wearable della Athos, vedi gure seguenti:

Figura 4.8: Athos wearable, [121]

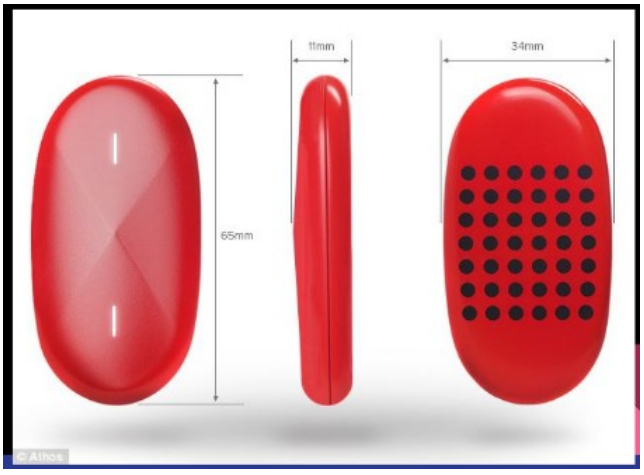


Figura 4.9: Athos package, [121]

Quindi, il package verrà collegato al wired-glove attraverso dei magneti, usati sia per l'ancoraggio del package all'interno dell'alloggiamento nel guanto, sia per la trasmissione dei segnali verso i sensori analogici (di bending e di forza) collocati sulle quattro dita del guanto nale. In questo caso, è stato preso spunto da un guanto realizzato in ambito gaming, ossia il Peregrine glove prodotto dalla Iron Will, [122]:

Figura 4.10: Peregrine glove, [122]

Pertanto, si tratta di un wired-glove innovativo per quanto riguarda l'ambito della riabilitazione medica. Infatti, tramite l'interfaccia graca, verrà controllato dall'utente nale per far svolgere al paziente gli opportuni trainings riabilitativi; garantendo un'acquisizione dati rapida ed ecente, con il loro conseguente invio, tramite bluetooth, al pc dove saranno analizzati in tempo realeLa struttura stessa del guanto è nuova, infatti una mini-PCB montata sul dorso del glove, con le tracce di rame integrate nel tessuto e-textile del guanto stesso non è presente nella letteratura analizzata.
In tal modo, dunque, si tenterà di favorire la riabilitazione e la conseguente guarigione dei pazienti che hanno subito gravi traumi, cercando d'introdurre un miglioramento nella corrente tecnologia riabilitativa.
La versione prototipale del package in cui verrà inserita la PCB miniaturizzata e dell'alloggiamento collocato sul dorso del guanto e in cui verrà disposto il package sono mostrati nelle seguenti gure:

Figura 4.11: Package aperto

Figura 4.12: Package chiuso

Nei tre "quadrati" presenti sulla supercie interna dell'alloggiamento, gura 4.11, saranno inseriti tre magneti per mantenere ferma ed in posizione la board progettata. Dopodichè, al di sopra della PCB e sempre, dunque, all'interno del package, verrà collocata la batteria LiPo che, servirà appunto per garantire l'alimentazione dell'intero circuito.

Capitolo 5

Capitolo 5: Programmazione della scheda prototipo

Dopo aver portato a termine il progetto della PCB miniaturizzata e aver completato la realizzazione della PCB, lo step sucessivo è rappresentato dalla programmazione della board per il testing e la verica del suo corretto funzionamento. Tuttavia, non è stata programmata direttamente la PCB miniaturizzata, bensì è stata sfruttata la scheda prototipo per il debugging del codice di programmazione.
Dunque, una volta completato l'assemblaggio della scheda miniaturizzata e del package, è stato realizzato un test preliminare del corretto funzionamento dei suoi sensori digitali. Per svolgere il testing è stata sfruttata la scheda-prototipo sulla quale sono stati montati gli stessi sensori, poi usati nella versione miniaturizzata. In particolare, è stato programmato il modulo nRF52832 usando il development kit PCA10040 versione 1.1.0 della Nordic Semiconductor, [109].

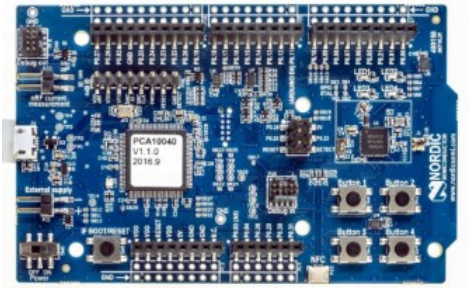


Figura 5.1: PCA10040 Development Kit

Come indicato nel capitolo 6 Hardware description della User Guide [110] del kit di sviluppo appena menzionato, è stato usato il connettore P20 di debug Out, specico per la programmazione di moduli della serie nRF52 posti al di fuori del "development kit" stesso.

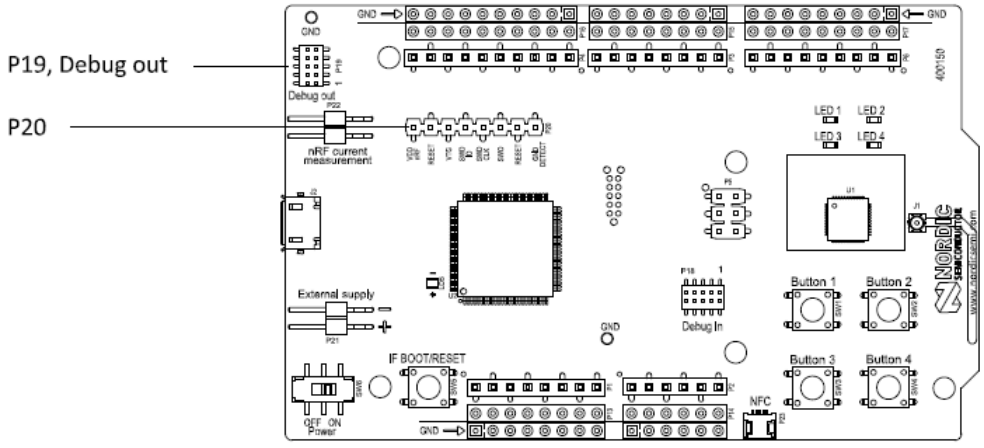


Figura 5.2: connettore P20/board pca10040, [111]

Tuttavia, prima di analizzare la parte hardware, è opportuno menzionare il software scelto per la programmazione iniziale della scheda-prototipo. Tornando indietro nella User Guide, si raggiunge il capitolo 3 Software tools, dove è elencato un gran numero di software usati per il testing e la programmazione dei chip della serie nRF52. Tra questi sono stati scelti quelli fondamentali per un corretto testing della scheda: l'nRF5 SDK o Software Development Kit, utile poichè fornisce esempi di programmi e contiene numerose librerie, che rappresentano la base per lo sviluppo del codice. Invece, come IDE di sviluppo è stato scelto SES (Segger Embedded Studio) versione 3.30 o successive [112]. La versione di SDK compatibile con SES è la "14.1" o successive e può essere scaricata al seguente link : https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF52832 (Products/Bluetooth low energy/nRF52832/Downloads). Invece, cliccando sul link presente nella User Guide (Capitolo 3/Software tools) si può scaricare la cosiddetta "documentation" relativa alla versione di SDK scelta in precedenza. In seguito, sono stati scaricati i cosiddetti J-link tools (https: //www.segger.com/downloads/jlink#J-LinkSoftwareAndDocumentationPack), molto utili nella fase di programmazione: ad esempio il J-link RTT Viewer può essere usato sia in fase di debugging sia in fase di running del programma, come monitor seriale (il funzionamento è lo stesso del Putty di Windows o del Serial monitor di Arduino). Inoltre, permette di vericare il corretto collegamento tra la board pca10040, alimentata dal pc tramite micro usb e la scheda-prototipo da testare.

Dopodichè, è stata realizzata una breve analisi della parte hardware della scheda di programmazione pca10040, iniziando dalle diverse sorgenti della tensione di alimentazione: connesione USB ; header a 2 pins maschio per la connessione ad una generica alimentazione esterna (range tensione 1.7 V - 3.6 V) e una batteria di

tipo coin cell per il funzionamento autonomo e senza cavo della scheda.

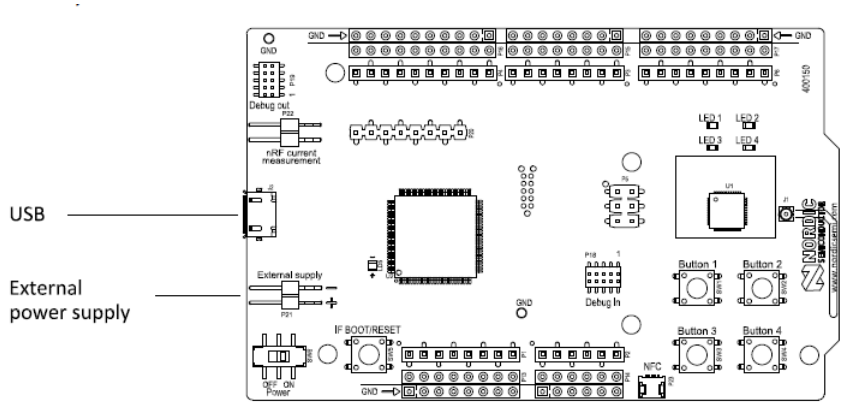


Figura 5.3: power supply top view, [111]

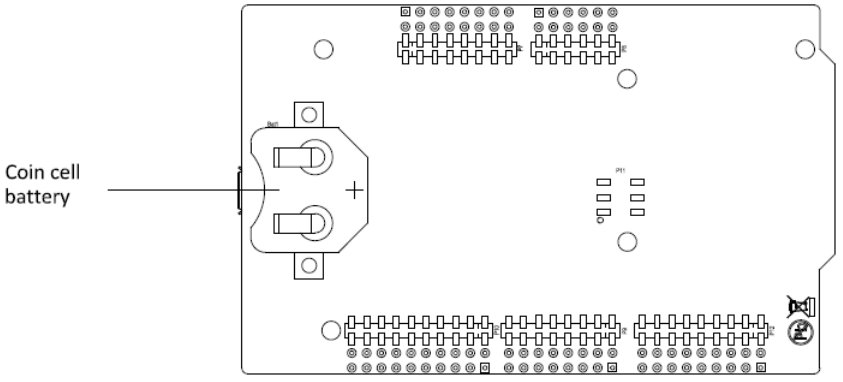


Figura 5.4: power supply bottom view, [111]

In particolare, la tensione proveniente dall'USB è di 5.0 V (come previsto dallo standard) è ridotta ad un valore di 3.3 V attraverso un regolatore di tensione interno; mentre le tensioni provenienti dall'header e dalla batteria non sono regolate. In prossimità dei tre ingressi delle tensioni di alimentazione sono stati inseriti tre diodi di protezione (D1A, D1B e D1C) per evitare malfunzionamenti o danneggiamenti sia della circuiteria interna della board sia della batteria o della sorgente di alimentazione esterna scelta. Uno tra i principali problemi può essere: errato montaggio della batteria che porterebbe ad un usso di corrente non più dalla batteria verso la pca10040, ma al contrario; provocando un reverse current owing, che conduce al danneggiamento della batteria stessa. Inserendo un diodo ciò può essere facilmente evitato, [113].
Lo svantaggio dell'inserimento dei diodi è rappresentato dalla caduta di tensione ai loro capi, che riduce la massima tensione fornita dalla sorgente di alimentazione

esterna. Per tale motivo, è possibile "bypassare" i diodi di protezione, cortocircuitando uno o più solder bridges (SB10, SB11, SB12) posti in parallelo ai diodi sopracitati.
Quanto descritto nora, è facilmente visibile nel seguente schematico:

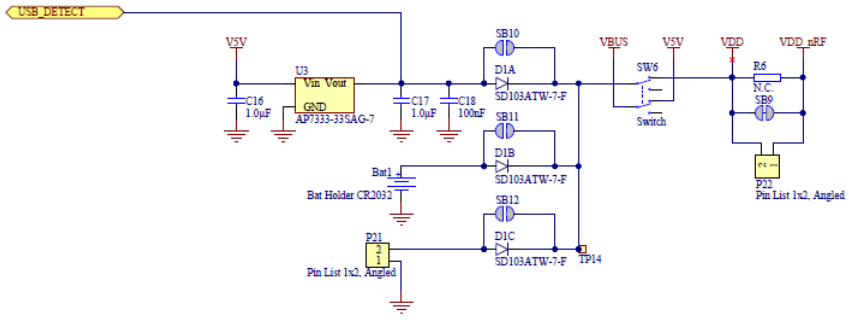


Figura 5.5: Schematico sezione di Power supply, [111]

L'altra parte importante riguardante l'hardware,interessa i connettori. Infatti, l'accesso ai pins di GPIO dell'nRF52832 è garantito attraverso i connettori P2, P3, P4, P5 e P6 ; mentre il connettore P1 permette di collegarsi alle zone di power e di ground della board di sviluppo Nordic, come mostrato nella seguente gura:

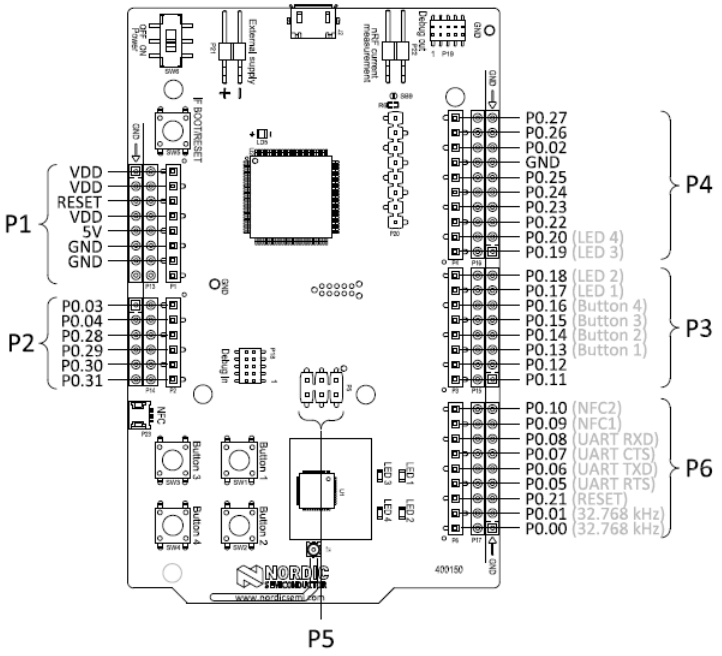


Figura 5.6: Connettori board pca10040, [111]

Inoltre, è possibile collegare la board pca10040 ad una scheda Arduino. I segnali provenienti dalla scheda Arduino sono collegati agli header di GPIO della board

Nordic come indicato nella gura seguente:

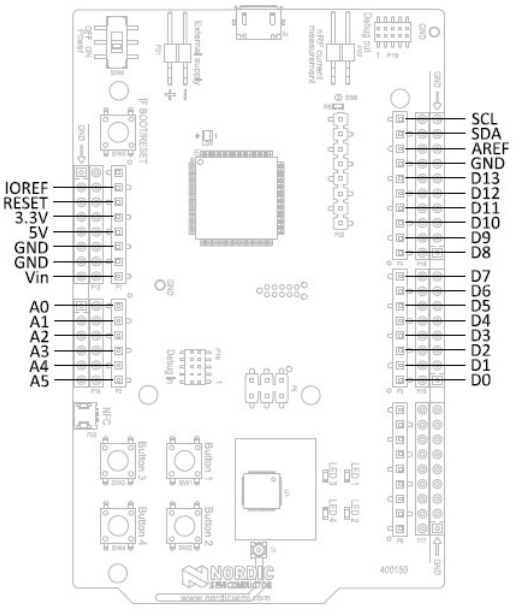


Figura 5.7: Connessione pca10040 con scheda Arduino, [111]

Pertanto, dopo aver scaricato il software SES(Segger Embedded Studio), l'evaluation board è stata alimentata dal pc tramite micro usb; quindi, la board di valutazione pca10040 e la scheda prototipo sono state connesse come mostrato nella seguente gura:

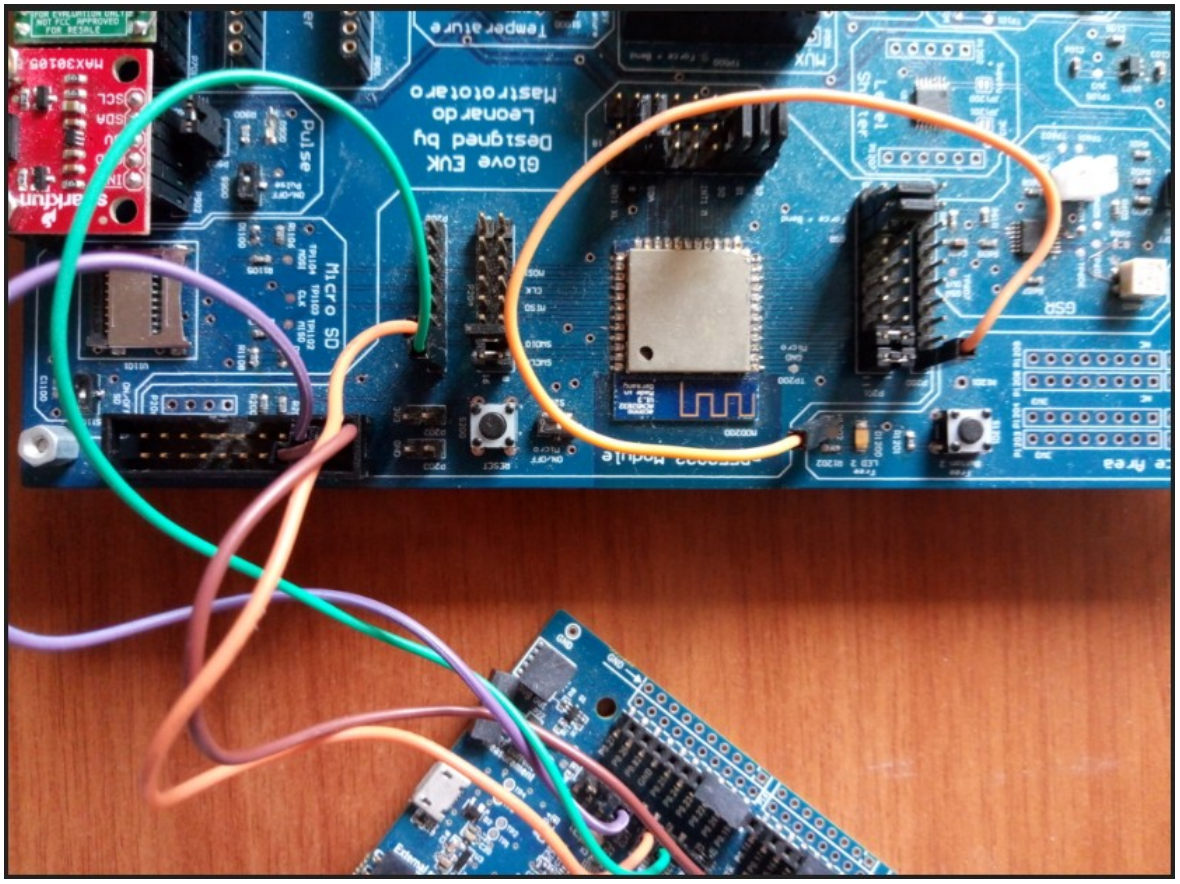


Figura 5.8: Connesssione della board pca10040 con la scheda prototipo

Come mostrato in gura 5.8, sono stati connessi quattro li jumper che trasportano rispettivamente i segnali di Vdd, GND, SWDIO e SWDCLK. Il quinto lo in gura 5.8, è stato usato per poter pilotare l'unico LED libero nella Service Area della scheda prototipo, utile nella fase di debugging del codice.
La parte di programmazione può essere suddivisa in tre parti principali: l'acquisizione dati da parte dei sensori digitali e analogici; la trasmissione dati attraverso il protocollo Bluetooth; la realizzazione dell'interfaccia graca per facilitare l'uso del wired-glove da parte dell'utente.In questo lavoro di tesi è stata realizzata la prima fase, ossia l'acquisizione dati da parte dei sensori digitali, connessi fra loro e al microcontrollore tramite il bus I2C.
Il primo passo, è stato l'analisi dei les d'esempio twi scanner pca10040 e twi sensor presenti nella libreria SDK scaricata.Infatti, il primo permette di eseguire uno scanning della board e stabilire quante periferiche sono connesse al bus I2C, fornendo inoltre i rispettivi address. Il secondo, invece, fornisce il codice necessario per l'acquisizione dati da un sensore di temperatura, che si basa sul protocollo seriale I2C per la loro comunicazione.Quindi, il secondo passo è stato scrivere il le Custom board.h ed includerlo nel le board.h, che specica, appunto, tutte le

board che racchiudono un microcontrollore della serie nRF52. Dopodichè, la restante parte del codice è stata scritta nel le main.c. Innanzitutto, è stata scritta la funzione board_scanning() per determinare il numero di sensori connessi all'I2C e stabilire se questi corrispondessero a quelli eettivamente montati sulla board, per vericare eventuali errori nelle connessioni della scheda prototipo. Infatti, è stato vericato che il sensore magnetometro LIS3MDL non è rilevato durante lo scanning, nonostante sia inserito sulla board e il suo corrispondente switch di accensione sia stato spostato su On. Di conseguenza, sono stati rilevati soltanto: il sensore di temperatura MAX30205 all'indirizzo 0x48, il pulsimetro MAX30105 all'indirizzo 0x57 e l'accelerometro-giroscopio 3D LSM6DS0 all'indirizzo 0x6a. Ciò conferma gli slave address riportati nei rispettivi datasheets. Quindi, analizzando i rispettivi datasheets, sono stati congurati i registri interni a ciascun sensore in modo da completare correttamente l'acquisizione dati. Le funzioni di setup sono le seguenti: MAX30205_set_mode(), MAX30105_power_up() e LSM6DS0_configuration(). Da sottolineare, che anche il codice relativo al magnetometro è stato scritto, in modo da averlo subito disponibile non appena risolto il problema sulla scheda prototipo; la funzione di congurazione del magnetometro è LIS3MDL_configuration().
All'interno della funzione board_scanning(), è stata denita la funzione valuta_sensore_on(), che stabilisce quale dei sensori digitali è attivo sulla board (ricordando, infatti, che possono essere disabilitati attraverso gli switch manuali) e memorizza l'indirizzo/gli indirizzi del/dei sensore/sensori attivi in quel momento. Dopodichè, sempre all'interno del main() del programma, è stato inserito il seguente ciclo:

```
_Delay(10);
user_display();
_Delay(10);
    char buer = 0;
    do{
    //buer = SEGGER_RTT_WaitKey();
    buer = getchar();
    read_input(buer);
    _Delay(10);
    user_display();
    _Delay(10);
    }
    while(buer!='e'); //'e to exit the execution of the program'
```

Il codice mostra come, viene mostrato a video un Menù Utente tramite la funzione user_display(). Da ricordare che, la funzione di monitor seriale è svolta

dall'RTT Viewer. Quindi, l'utente inserirà uno dei caratteri indicati nel menù, in modo da acquisire i dati dal sensore che è stato selezionato tramite tale carattere. La funzione read_input() sceglierà il sensore corretto, in base al carattere, ricevuto come parametro d'ingresso e verrà di conseguenza svolta la lettura dei dati. Tale procedimento, può essere ripetuto dall'utente per svolgere acquisizioni in istanti di tempo diversi, sino a che non verrà inserito il carattere 'e' che terminerà l'esecuzione del programmall codice completo, relativo al debugging dei sensori digitali presenti sulla scheda prototipo, è stato riportato completamente nell'Appendice A di questa tesi.

Appendici

Appendice A

Codice C++ di programmazione

A.1 Custom_board.h

```
#ifndef CUSTOM_BOARD_H_INCLUDED
#dene CUSTOM_BOARD_H_INCLUDED

#ifdef __cplusplus
extern "C" {
#endif

#include "nrf_gpio.h"

#endif // CUSTOM_BOARD_H_INCLUDED

// denizione di SDA e SCL
#dene ACONNO_MODULE_SDA 12 //PO.12
#dene ACONNO_MODULE_SCL 11 //PO.11

// denizione SPI pins
#dene ACONNO_MODULE_MOSI 17 //PO.17
#dene ACONNO_MODULE_MISO 19 //PO.19
#dene ACONNO_MODULE_CLK 18 //PO.18
#dene ACONNO_MODULE_CS 20 //PO.20

// denizione pins SWDIO e SWDCLK
/* #dene ACONNO_MODULE_SWDIO SWDIO //SWDIO
#dene ACONNO_MODULE_SWDCLK SWDCLK //SWDCLK */
```

```
//denizione segnali multiplexer
#dene ACONNO_MODULE_S0 7 //PO.07
#dene ACONNO_MODULE_S1 6 //PO.06
#dene ACONNO_MODULE_S2 5 //PO.05
#dene MUX_OUTPUT 3 //PO.03

// denizione LEDs per la CUSTOM_BOARD
#dene LEDS_NUMBER 1

#dene LED_START 25 //PO.25
#dene LED_1 25 //PO.25
#dene LED_STOP 25 //PO.25

#dene LEDS_ACTIVE_STATE 0

// Low frequency clock source to be used by the SoftDevice
#dene NRF_CLOCK_LFCLKSRC {.source =
      NRF_CLOCK_LF_SRC_XTAL, \
.rc_ctiv = 0, \
.rc_temp_ctiv = 0, \
.accuracy = NRF_CLOCK_LF_ACCURACY_20_PPM}

#ifdef __cplusplus
}
#endif

#endif // CUSTOMBOARD_H
```

A.2 main.c

A.2.1 main()

```
 // ************* inizializzazione ******************

//inizializzazione del led 2 della "Service Area" della board di "L.Mastrototaro"
bsp_board_init(BSP_INIT_LEDS);
/* ret_code_t err_code;
uint8_t address;
uint8_t sample_data;
bool detected_device = false;*/
```

```
APP_ERROR_CHECK(NRF_LOG_INIT(NULL));
NRF_LOG_DEFAULT_BACKENDS_INIT();

//SEGGER_RTT_Init();
SEGGER_RTT_SetTerminal(0);
twi_init();

//**************** scanning della board*****************************

board_scanning();

//*****************congurazione dei sensori************************

/*scambio dati su TWI (I2C) con il sensore di temperatura
il sensore di temperatura si trova all'indirizzo 0x48 */
twi_handle_events();

if (temp_on)
{
MAX30205_set_mode(); //per la prima misurazione di temperatura
}

if (pulse_oxim_on)
{
//sensore MAX30105 pulse¯ oximeter si trova all'indirizzo 0x57
MAX30105_power_up();
//MAX30105_particlesensing_2LEDs();
}

if (acc_gyr_on)
{
LSM6DS0_conguration();
}

if (magnet_on)
{
LIS3MDL_conguration();
}

//************ Menu utente ************************
_Delay(10);
```

```
user_display();
_Delay(10);


char buer = 0;
do{
//buer = SEGGER_RTT_WaitKey();
buer = getchar();
read_input(buer);
_Delay(10);
user_display();
_Delay(10);
}
while(buer!='e'); //'e to exit the execution of the program'

}
```

A.2.2 Board_scanning()

```
void board_scanning()
{
ret_code_t err_code;
uint8_t address;
uint8_t sample_data;
bool detected_device = false;

int contaTWI = 0, slave_index = 0, somma = 0;

printf("Inizio scansione della board: ricerca sensori connessi all'I2C \n\n\n");

while ((detected_device != true) && (contaTWI <= 4))
{
contaTWI = 0;
//printf("TWI scanner start...\n");
SEGGER_RTT_printf(0,"TWI scanner start...\n");
_Delay(50);
for (address = 0; address <= TWI_ADDRESSES; address++)
{
err_code = nrf_drv_twi_rx(&m_twi, address, &sample_data, sizeof(
    sample_data));
//printf("Error code : %d\n", err_code);
```

```
if (err_code == NRF_SUCCESS) {
detected_device = true;
contaTWI++;
slave_index++;
nrf_delay_ms(400);
//printf("%d: TWI device detected at address 0x%x.\n", address, address);
SEGGER_RTT_printf(0,"%d: TWI device detected at address 0x%x.\n",
    address, address);
_Delay(50);
//salvo gli indirizzi trovati in un vettore per poterli usare in seguito
address_vector[slave_index] = address;
// faccio il toggle del LED 2
bsp_board_led_invert(0);
nrf_delay_ms(500);
}
// NRF_LOG_FLUSH();
}
if ((!detected_device) || contaTWI == 0) {
//printf("No device was found.\n\n");
SEGGER_RTT_printf(0,"No device was found.\n\n");
_Delay(50);
//NRF_LOG_FLUSH();
} else {
//printf("Ho trovato %d dispositivi.\n\n", contaTWI);
SEGGER_RTT_printf(0,"Ho trovato %d dispositivi.\n\n", contaTWI);
_Delay(50);
//NRF_LOG_INFO("Ho trovato %d dispositivi.\n\n", contaTWI);
//NRF_LOG_FLUSH();
valuta_sensore_on(contaTWI, address_vector);
}

nrf_delay_ms(500);

}

}
```

A.2.3 Congurazione e inizializzazioni

```
/* TWI instance ID. */
#if TWI0_ENABLED
#dene TWI_INSTANCE_ID 0
```

```
#elif TWI1_ENABLED
#dene TWI_INSTANCE_ID 1
#endif

#ifndef NRF_LOG_USES_RTT
#dene NRF_LOG_USES_RTT 1
#endif


#dene PI 3.141
/*denizione di Tmax e Tmin (scelte da me, possono essere cambiate)*/
#dene Tmax 0x28U
#dene Tmin 0x01U

// MASSIMO NUMERO DI SAMPLES all'interno della memoria FIFO del
    pulse¯oximeter
#dene MAX_SAMPLES_FIFO 32

/*denizione dei Pointer bytes dei 4 registri del temp sensor MAX30205 */
#dene MAX30205_REG_TEMP 0x00U
#dene MAX30205_REG_CONF 0x01U
#dene MAX30205_REG_THYST 0x02U
#dene MAX30205_REG_TOS 0x03U

/*denizione dei register addresses per i registri del pulse¯oximeter sensor
    MAX30105 */
//status registers
#dene MAX30105_INTERRUPTSTATUS1 0x00U
#dene MAX30105_INTERRUPTSTATUS2 0X01U
//FIFO conguration registers
#dene MAX30105_FIFOCONF 0x08U
#dene MAX30105_FIFOWRPTR 0x04U //FIFO write pointer register
#dene MAX30105_FIFORDPTR 0x06U //FIFO read pointer register
#dene MAX30105_FIFO_DATA 0x07U //FIFO data register
//conguration registers
#dene MAX30105_INTERRUPTENABLE1 0x02U
#dene MAX30105_MODECONF 0x09U
#dene MAX30105_PARTICLESENS_CONF 0x0AU
#dene MAX30105_LEDPULSEAMPL_LED1 0x0CU
#dene MAX30105_LEDPULSEAMPL_LED2 0x0DU
#dene MAX30105_LEDPULSEAMPL_PROXMODE 0x10U
```

```
#dene MAX30105_PROXMODE_INT_THRESH 0x30U
//temperature registers
#dene MAX30105_TEMPINT 0x1FU //valore intero della temperatura
#dene MAX30105_TEMPFRAC 0x20U //valore decimale della temperatura
#dene MAX30105_DIETEMP_CONF 0x21U

/*denizione dei register addresses per i registri dell' iNemo LSM6DSO */
//FIFO conguration registers
#dene LSM6DS0_FIFO_CTRL 0x2EU
#dene LSM6DS0_FIFO_SRC 0X2FU
//interrupt control registers
#dene LSM6DS0_INT_CTRL 0X0CU
//control registers
#dene LSM6DS0_CTRL_REG1_G 0x10U
#dene LSM6DS0_CTRL_REG4 0x1EU
#dene LSM6DS0_CTRL_REG5_XL 0x1FU
#dene LSM6DS0_CTRL_REG6_XL 0x20U
#dene LSM6DS0_CTRL_REG8 0X22U
#dene LSM6DS0_CTRL_REG9 0x23U
//status registers
#dene LSM6DS0_STATUS_REG_17 0X17U
#dene LSM6DS0_STATUS_REG_27 0X27U
//gyroscope output data registers
#dene LSM6DS0_OUT_X_L_G 0x18U
#dene LSM6DS0_OUT_X_H_G 0x19U
#dene LSM6DS0_OUT_Y_L_G 0x1AU
#dene LSM6DS0_OUT_Y_H_G 0x1BU
#dene LSM6DS0_OUT_Z_L_G 0x1CU
#dene LSM6DS0_OUT_Z_H_G 0x1DU
//accelerometer output data registers
#dene LSM6DS0_OUT_X_L_XL 0x28U
#dene LSM6DS0_OUT_X_H_XL 0x29U
#dene LSM6DS0_OUT_Y_L_XL 0x2AU
#dene LSM6DS0_OUT_Y_H_XL 0x2BU
#dene LSM6DS0_OUT_Z_L_XL 0x2CU
#dene LSM6DS0_OUT_Z_H_XL 0x2DU

/*denizione dei register addresses per i registri del magnetometro LIS3MDL */
//control registers
#dene LIS3MDL_CTRL_REG1 0x20U
#dene LIS3MDL_CTRL_REG2 0x21U
```

```
#dene LIS3MDL_CTRL_REG3 0x22U
#dene LIS3MDL_CTRL_REG4 0x23U
#dene LIS3MDL_CTRL_REG5 0x24U
//status registers
#dene LIS3MDL_STATUS_REG 0x27U
//output data registers
#dene LIS3MDL_OUT_X_L 0x28U
#dene LIS3MDL_OUT_X_H 0x29U
#dene LIS3MDL_OUT_Y_L 0x2AU
#dene LIS3MDL_OUT_Y_H 0x2BU
#dene LIS3MDL_OUT_Z_L 0x2CU
#dene LIS3MDL_OUT_Z_H 0x2DU

/*congurazione SCELTA DA ME per il funzionamento del sensore
    magnetometro LIS3MDL considerando attive solo asse X e asse Y*/
#dene TEMP_EN_M 0x00U // 0 non abilito il sensore di temperatura interno
     al magnetometro
#dene OM 0x20U //01 operating mode X,Y axis medium performance
#dene DO 0x18U //110 ODR = 40 Hz
#dene FAST_ODR 0x00U //0 disabilito il fast ODR
#dene FS 0x40U //10 full scale impostato a +/    − 12 [gauss]
#dene REBOOT 0X00U // 0
#dene SOFT_RST 0x04U // 1 software reset all'inizio della congurazione
#dene LP 0x00U // 0 Low power mode disabilitata
#dene SIM 0X00U // 0
#dene MD 0x00U // 00 operating mode selection : scelto la "continous  −
    conversion mode"
#dene OMZ 0x00U // 00 Z   − axis operative mode : disabled
#dene FAST_READ 0x80U // 1 abilito il FAST READ
#dene BDU_m 0x00U // 0 block data update disabilitato

/*congurazione SCELTA DA ME per il funzionamento del sensore iNEMO
    LSM6DS0
considero attivi entrambi accelerometro 3D e giroscopio 3D (si veda datasheet)
    */
#dene FMODE 0x00U // 000 usati per settare la Bypass mode (FiFO is
    turned o)
#dene FTH 0x00U // 00000 impostano la FIFO threshold, che rappresenta la "
    depth" della FIFO
#dene STOP_ON_FTH 0x01U // 1 abilita la denizione di una "threshold"
    per la FIFO depth
```

```
#dene FIFO_EN 0x01U // 1 bit per abilitare la FIFO
#dene INT_FTH 0x01U
#dene ODR_G 0xA0U // 101 setto l'output data rate, quando sia gyroscope
    sia accelerometer sono attivi (476 Hz)
#dene FS_G 0x08U // 01 setto il FS(Full Scale) dell'accelerometro pari a 500
    dps
#dene ODR_XL 0x80U // 100
#dene FS_XL 0x08U // 01 full scale dell'accelerometro pari a +    ⁻16g
#dene BDU 0x00U // 0 block data update
#dene IF_ADD_INC 0x04U // 1
#dene SW_RESET 0x01U // 1 software reset
#dene SLEEP_G 0x00U // 0 abilito il giroscopio

/*congurazione SCELTA DA ME per il funzionamento in
PARTICLE ⁻ SENSING MODE con 2 LEDs per il sensore MAX30105 (si veda
    datasheet) */
#dene PROX_INT_EN 0x00U // 0
#dene SMP_AVE 0x00U // 000
#dene FIFO_ROLLOVER_EN 0x00U // 0
#dene FIFO_A_FULL 0x0CU // 1100 triggero l'interrupt nella fo quando ho
     raccolto 20 samples
#dene SHDN 0x00U // 0
#dene RESET 0x01U // 1
#dene MODE 0x03U // 011 per ATTIVARE il funzionamento con 2 LEDs
#dene ADC_RGE 0x01U // 01
#dene SR 0x00U // 000
#dene LED_PW 0x03U // 11
#dene LEDx_PA 0x0FU //valido sia per il LED1 che per il LED2 (
    corrisponde ad un valore di corrente di 3.1 mA)
#dene TEMP_EN 0x01U // 1 abilito il sensore di temperatura interno al
    MAX30105
#dene PROX_INT_THRESH 0x01U //sono gli 8 MSBs dell'ADC count

/*Congurazione SCELTA DA ME per il sensore MAX30205 (si veda datasheet)
     */
#dene shutdown_modeD0 1 //sensore MAX30205 funzionamento in SHUT
    DOWN mode
#dene OScomparator_modeD1 0 //uscita OS in COMPARATOR MODE
#dene OSpolarityD2 0
#dene D3 1
#dene D4 0
```

```
#dene normal_dataD5 0 //formato dei valori di temperatura NORMAL DATA
      (nei registri TOS e THYST) su 16 bits in 2'C
#dene bus_timeoutD6 0 // abilito il bus time      ⁻ out
#dene one_shotD7 1 //abilito la ONE     ⁻ SHOT conversion

#dene ADDR_NUMBER 4 //numero d'indirizzi pari al numero di sensori
     digitali sulla scheda prototipo
#dene PRECISION 1000

/*indirizzi dei 4 sensori digitali presenti sulla board di L.Mastrototaro */
uint8_t temp_sensor_address = 0x00U;
uint8_t pulse_ox_address = 0x00U;
uint8_t accel_gyr_address = 0x00U;
uint8_t magnetometer_address = 0x00U;

/* Number of possible TWI addresses. */
// #dene TWI_ADDRESSES 127
#dene TWI_ADDRESSES 180 // relativi a : sensore di temperatura,
     accelerometro + giroscopio 3D, magnetometro 3D e sensore pulsoxiimeter


/* TWI instance. */
static const nrf_drv_twi_t m_twi = NRF_DRV_TWI_INSTANCE(
     TWI_INSTANCE_ID);

/* Indicates if operation on TWI has ended. */
static volatile bool m_xfer_done = false;

oat temp_decimal;

/* Buer for samples read from slave sensors. */
static uint8_t m_sample;

static uint8_t temp_int;
static uint8_t temp_frac;
static uint8_t FIFO_rdptr, FIFO_wrptr;
static uint8_t FIFO_data;
//static uint8_t interrupt_status1_value;
static uint8_t ir_data_23_16 = 0, ir_data_15_8 = 0, ir_data_7_0 = 0;

static int16_t OUT_X_G, OUT_Y_G, OUT_Z_G; //variabili dove salvare
```

```
    le uscite lungo x,y, e z del giroscopio (LSM6DSO)
static int16_t OUT_X_XL, OUT_Y_XL, OUT_Z_XL; // variabili dove
    salvare le uscite lungo x,y, e z dell'accelerometro (LSM6DSO)
static int16_t OUT_X, OUT_Y, OUT_Z; //variabili dove salvare le uscite
    lungo x,y, e z del magnetometro (LIS3MDL)

/* Address of the TWI devices detected on the "L.Mastrototaro" board */
static uint8_t address_vector[ADDR_NUMBER];


//LSM6DS0 accelerometer/gyroscope
oat G_sensitivity = 17.5e  - 3; // gyroscope sensitivity [dps/LSB] con FS =
    +/ - 500 dps
oat LA_sensitivity = 0.732e  - 3; //linear acceleration sensitivity [g/LSB] con
    FS = +/ - 16 [g]
int GN_sensitivity = 2281; //magnetometer sensitivity [LSB/gauss] con FS =
    +/ - 12 [gauss]
static int16_t oset_asseX, oset_asseY, oset_asseZ;
oat asseX_LSB = 0, asseY_LSB = 0, asseZ_LSB = 0;
bool X_axis, Z_axis, Y_axis;


bool temp_on = false;
bool pulse_oxim_on = false;
bool acc_gyr_on = false;
bool magnet_on = false;
bool t = false,p = false,a = false,g = false,m = false;
```

A.2.4 Funzioni congurazione sensori

```
/**
* @brief Function for handling data from temperature sensor.
*
* @param[in] temp Temperature in Celsius degrees read from sensor.
*/

__STATIC_INLINE void data_handler_rx(uint8_t temp1, uint8_t temp2,
    oat temp3) {
if (t)
{
printf("Temperature: %d Celsius degrees. \n", temp1);
}
```

```
if (p)
{
printf("Temp misurata con il pulse − oximeter: %d.%3d Celsius degrees. \n ",
      temp2, (int)(temp3*1000));
}

}

__STATIC_INLINE void data_handler_tx(void) {
printf("Congurazione registri terminata con successo \n");
}

//SEGGER delay
static void _Delay(int period)
{
int i = 100000*period;
do{;} while(i    −− );
}

/**
* @brief TWI events handler.
*/
void twi_handler(nrf_drv_twi_evt_t const *p_event, void *p_context) {
switch (p_event  − >type) {
case NRF_DRV_TWI_EVT_DONE:
if (p_event − >xfer_desc.type == NRF_DRV_TWI_XFER_RX) {
data_handler_rx(m_sample, temp_int, temp_decimal);
}
m_xfer_done = true;
break;
if (p_event − >xfer_desc.type == NRF_DRV_TWI_XFER_TX) {
data_handler_tx();
}
m_xfer_done = true;
default:
break;
}
}

/**
```

```
* @brief TWI initialization.
*/
void twi_init(void) {
ret_code_t err_code;

/* const nrf_drv_twi_cong_t twi_cong = {
.scl = ARDUINO_SCL_PIN,
.sda = ARDUINO_SDA_PIN,
.frequency = NRF_DRV_TWI_FREQ_100K,
.interrupt_priority = APP_IRQ_PRIORITY_HIGH,
.clear_bus_init = false */

// congurazione TWI per la Custom board
const nrf_drv_twi_cong_t twi_cong = {
.scl = ACONNO_MODULE_SCL,
.sda = ACONNO_MODULE_SDA,
.frequency = NRF_DRV_TWI_FREQ_100K, // frequenza trasmissione dati
      100 kbps
.interrupt_priority = APP_IRQ_PRIORITY_HIGH,
.clear_bus_init = false

};

err_code = nrf_drv_twi_init(&m_twi, &twi_cong, NULL, NULL);
printf("Error code : %d \n", err_code);

APP_ERROR_CHECK(err_code);

nrf_drv_twi_enable(&m_twi);
}

/**
* @brief Function used to handle TWI events
*/

void twi_handle_events(void) {
ret_code_t err_code;
nrf_drv_twi_uninit(&m_twi);

// congurazione TWI per la Custom board
const nrf_drv_twi_cong_t twi_cong = {
```

```
.scl = ACONNO_MODULE_SCL,
.sda = ACONNO_MODULE_SDA,
.frequency = NRF_DRV_TWI_FREQ_100K, // frequenza trasmissione dati
    100 kbps
.interrupt_priority = APP_IRQ_PRIORITY_HIGH,
.clear_bus_init = TWI_DEFAULT_CONFIG_CLR_BUS_INIT //clear bus
    during initialization

};

err_code = nrf_drv_twi_init(&m_twi, &twi_cong, twi_handler, NULL);
printf("Error code : %d \n", err_code);

APP_ERROR_CHECK(err_code);

nrf_drv_twi_enable(&m_twi);
}
/**
* @brief Function used to establish the correct power up of the MAX30105
    pulse oximeter sensor
*/

void MAX30105_power_up() {

ret_code_t err_code;
//conguro il DIE TEMPERATURE CONFIG register
m_xfer_done = false;

uint8_t temp_conf_byte = 0x00U;
temp_conf_byte = temp_conf_byte ^ TEMP_EN;
uint8_t conf_DIETEMP[2] = {MAX30105_DIETEMP_CONF,
    temp_conf_byte};

err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address, conf_DIETEMP,
    sizeof(conf_DIETEMP), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

//lettura dello status register "Interrupt Status 1 "
//ret_code_t err_code;
```

```
m_xfer_done = false;

printf("Indirizzo sensore pulse- oximeter: 0x%x. \n", pulse_ox_address);
/* Leggo 1 byte di dati alla volta, all'indirizzo specicato per il sensore PULSE -
    OXIMETER */

uint8_t value = MAX30105_INTERRUPTSTATUS1;

//1^a fase la TX : invio di slave address + pointer byte da parte del master
    verso il sensore pulse oximeter (slave)
err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address, &value, 1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

m_xfer_done = false;

//2^a fase la RX: nuovo start da parte del master; successivo invio dello slave
    address , e a questo punto si possono leggere i data bytes dal
// registro dello slave
err_code = nrf_drv_twi_rx(&m_twi, pulse_ox_address, &m_sample, sizeof(
    m_sample));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

int pwr_rdy;
pwr_rdy = m_sample;

if (pwr_rdy == 0){
printf("Pwr_rdy = %d \n\n Sensore MAX30105      pronto per il funzionamento \
    n", pwr_rdy);
}
else
printf("Pwr_rdy = %d ; Sensore MAX30105 non alimentato correttamente. \n
    Connettere all'alimentazione !!! \n\n", pwr_rdy);

nrf_delay_ms(500);
}
```

```
/**
* @brief Function used to congure the LIS3MDL magnetometer sensor
*/
void LIS3MDL_conguration()
{
ret_code_t err_code;
m_xfer_done = false;
//magnetometer_address = 0x38U;

nrf_delay_ms(300);
printf("Indirizzo del sensore LIS3MDL: 0x%x. \n ", magnetometer_address);

//conguro il CTRL_REG2 register (per resettare il dispositivo)
m_xfer_done = false;
nrf_delay_ms(200);

uint8_t ctrl2_byte = (FS^REBOOT)^SOFT_RST; // 01000100 = 0X44
uint8_t conf_CTRL2[2] = {LIS3MDL_CTRL_REG2, ctrl2_byte};

err_code = nrf_drv_twi_tx(&m_twi, magnetometer_address, conf_CTRL2,
    sizeof(conf_CTRL2), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

//conguro il CTRL_REG1 register
m_xfer_done = false;
nrf_delay_ms(200);

uint8_t ctrl1_byte = ((TEMP_EN_M^OM)^DO)^FAST_ODR; // 00111000
    = 0x38
uint8_t conf_CTRL1[2] = {LIS3MDL_CTRL_REG1, ctrl1_byte};

err_code = nrf_drv_twi_tx(&m_twi, magnetometer_address, conf_CTRL1,
    sizeof(conf_CTRL1), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);
```

```
//conguro il CTRL_REG3 register
m_xfer_done = false;
nrf_delay_ms(200);

uint8_t ctrl3_byte = (LP^SIM)^MD; //00000000 = 0x00
uint8_t conf_CTRL3[2] = {LIS3MDL_CTRL_REG3, ctrl3_byte};

err_code = nrf_drv_twi_tx(&m_twi, magnetometer_address, conf_CTRL3,
    sizeof(conf_CTRL3), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

//conguro il CTRL_REG4 register
m_xfer_done = false;
nrf_delay_ms(200);

uint8_t ctrl4_byte = OMZ; //00000000 = 0x00
uint8_t conf_CTRL4[2] = {LIS3MDL_CTRL_REG4, ctrl4_byte};

err_code = nrf_drv_twi_tx(&m_twi, magnetometer_address, conf_CTRL4,
    sizeof(conf_CTRL4), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

//conguro il CTRL_REG5 register
m_xfer_done = false;
nrf_delay_ms(200);

uint8_t ctrl5_byte = FAST_READ^BDU_m; //10000000 = 0x80
uint8_t conf_CTRL5[2] = {LIS3MDL_CTRL_REG5, ctrl5_byte};

err_code = nrf_drv_twi_tx(&m_twi, magnetometer_address, conf_CTRL5,
    sizeof(conf_CTRL5), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);


}
```

```
/**
* @brief Function used to congure the LSM6DSO sensor: both accelerometer
      and gyroscope are enabled
*/
void LSM6DS0_conguration()
{
ret_code_t err_code;
m_xfer_done = false;

nrf_delay_ms(300);
printf("Indirizzo del sensore iNemo LSM6DS0: 0x%x. \n ", accel_gyr_address);

//conguro il CTRL_REG8 register (per resettare il dispositivo)
m_xfer_done = false;
nrf_delay_ms(200);

uint8_t ctrl8_byte = (BDU^IF_ADD_INC)^SW_RESET; // 00000101 = 0
      x05
uint8_t conf_CTRL8[2] = {LSM6DS0_CTRL_REG8, ctrl8_byte};

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, conf_CTRL8,
      sizeof(conf_CTRL8), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);


//conguro il FIFO_CTRL register

uint8_t fo_byte = 0x00U;
//fo_byte = (fo_byte + (FMODE<<5)) + FTH ; // il risultato di queste
      operazioni deve essere 00111111 = 0x3F
uint8_t conf_FIFOCTRL[2] = {LSM6DS0_FIFO_CTRL, fo_byte};


err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, conf_FIFOCTRL,
      sizeof(conf_FIFOCTRL), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
```

```
while (m_xfer_done == false);

//conguro il CTRL_REG1_G register
m_xfer_done = false;

nrf_delay_ms(200);

uint8_t ctrl1_byte = 0x00U;
ctrl1_byte = (ctrl1_byte^ODR_G)^FS_G; //il risultato di queste operazioni
    deve essere 10101000 = 0xA8U
uint8_t conf_CTRL1[2]= {LSM6DS0_CTRL_REG1_G, ctrl1_byte};

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, conf_CTRL1,
    sizeof(conf_CTRL1), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

//conguro il CTRL_REG4 register
m_xfer_done = false;

nrf_delay_ms(200);

uint8_t ctrl4_byte = 0x38U; // 00111000
uint8_t conf_CTRL4[2]= {LSM6DS0_CTRL_REG4, ctrl4_byte};

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, conf_CTRL4,
    sizeof(conf_CTRL4), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

//conguro il CTRL_REG5_XL register
m_xfer_done = false;

nrf_delay_ms(200);

uint8_t ctrl5_byte = 0x38U; //00111000
uint8_t conf_CTRL5[2] = {LSM6DS0_CTRL_REG5_XL, ctrl5_byte};

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, conf_CTRL5,
```

```
    sizeof(conf_CTRL5), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

//conguro il CTRL_REG6_XL register
m_xfer_done = false;

nrf_delay_ms(200);

uint8_t ctrl6_byte = ODR_XL^FS_XL; // 10001000 = 0x88
uint8_t conf_CTRL6[2] = {LSM6DS0_CTRL_REG6_XL, ctrl6_byte};

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, conf_CTRL6,
    sizeof(conf_CTRL6), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);



//conguro il CTRL_REG9 register
m_xfer_done = false;

nrf_delay_ms(200);

uint8_t ctrl9_byte = SLEEP_G; //il risultato di queste operazioni deve essere
    01000000 = 0x40
uint8_t conf_CTRL9[2]= {LSM6DS0_CTRL_REG9, ctrl9_byte};

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, conf_CTRL9,
    sizeof(conf_CTRL9), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

//conguro l'INT_CTRL register
m_xfer_done = false;

nrf_delay_ms(200);
```

```
uint8_t int_ctrl_byte = 0x00U;
int_ctrl_byte = (int_ctrl_byte + (INT_FTH<<3)); //il risultato di queste
      operazioni deve essere 00001000 = 0x08

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &int_ctrl_byte,
      sizeof(int_ctrl_byte), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);


nrf_delay_ms(500);
}

/**
* @brief Function used to enter into PARTICLE SENSING MODE and initiate
      a temperature measurement with the MAX30105 pulse oximeter sensor
*/

void MAX30105_particlesensing_2LEDs() {
ret_code_t err_code;
m_xfer_done = false;

/*
//conguro il MODE CONFIGURATION register

uint8_t mode_byte = SHDN;
mode_byte = ((mode_byte ^ RESET) << 6) ^ MODE; // il risultato di queste
       operazioni deve essere 01000011 = 0x43
uint8_t conf_MODEREG[2] = {MAX30105_MODECONF, mode_byte};

err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address, conf_MODEREG,
      sizeof(conf_MODEREG), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;
*/

// conguro il PARTICLE SENSING CONFIGURATION register
m_xfer_done = false;
```

```
uint8_t particle_sensing_byte = SR;
particle_sensing_byte = ((particle_sensing_byte ^ ADC_RGE) << 5) ^
    LED_PW; // il risultato di queste operazioni dev'essere 00100011 = 0x23
uint8_t conf_PARTSENS[2] = {MAX30105_PARTICLESENS_CONF,
    particle_sensing_byte};

err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address, conf_PARTSENS,
    sizeof(conf_PARTSENS), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;


// conguro il LED PULSE AMPLITUDE register (LED 1)
m_xfer_done = false;

uint8_t conf_LEDPULSE1[2] = {MAX30105_LEDPULSEAMPL_LED1,
    LEDx_PA};

err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address, conf_LEDPULSE1,
    sizeof(conf_LEDPULSE1), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

// conguro il LED PULSE AMPLITUDE register (LED 2)
m_xfer_done = false;

uint8_t conf_LEDPULSE2[2] = {MAX30105_LEDPULSEAMPL_LED2,
    LEDx_PA};

err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address, conf_LEDPULSE2,
    sizeof(conf_LEDPULSE2), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;
/*
```

```
//conguro il DIE TEMPERATURE CONFIG register
m_xfer_done = false;

uint8_t temp_conf_byte = 0x00U;
temp_conf_byte = temp_conf_byte ^ TEMP_EN;
uint8_t conf_DIETEMP[2] = {MAX30105_DIETEMP_CONF,
    temp_conf_byte};

err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address, conf_DIETEMP,
    sizeof(conf_DIETEMP), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);*/

nrf_delay_ms(500);
}

/**
* @brief Function used to set the MAX30105 in Proximity Mode
*/
void MAX30105_PROXIMITY_MODE(){
ret_code_t err_code;
uint8_t IR_data[3] = {0,0,0}, vettore = 0x00U;
bool nito = false;
int i = 0, ir_value = 0;

printf("\n\n Appoggiare il dito sul pulsimetro... \n\n");
nrf_delay_ms(20);
//MODE conguration register
m_xfer_done = false;

uint8_t mode_byte = 0x43U; //01000011
uint8_t mode_conf[2] = {MAX30105_MODECONF, mode_byte};

err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address, mode_conf, sizeof(
    mode_conf), false);
printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;
```

```
//Proximity mode LED pulse Amplitude register
m_xfer_done = false;

uint8_t Pilot_PA = 0xFFU;
uint8_t prox_mode_led_conf[2] = {
    MAX30105_LEDPULSEAMPL_PROXMODE, Pilot_PA};

err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address,
    prox_mode_led_conf, sizeof(prox_mode_led_conf), false);
printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

//conguro il FIFO conf register
m_xfer_done = false;

uint8_t FIFO_conf_byte = FIFO_ROLLOVER_EN;
FIFO_conf_byte = ((FIFO_conf_byte ^ SMP_AVE) << 5) ^
    FIFO_A_FULL; //il risultato di queste operazioni dev'essere
uint8_t conf_FIFO[2] = {MAX30105_FIFOCONF, FIFO_conf_byte};

err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address, conf_FIFO, sizeof(
    conf_FIFO), false);
printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;


//Proximity mode Interrupt Threshold register
m_xfer_done = false;

uint8_t Prox_int_thresh = 0x01U;// soglia dell'IR ADC pari a 1023 in
    decimale
uint8_t prox_mode_int_thr_conf[2] = {
    MAX30105_PROXMODE_INT_THRESH, Prox_int_thresh};

err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address,
```

```
    prox_mode_int_thr_conf, sizeof(prox_mode_int_thr_conf), false);
printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

//conguro l'INTERRUPT ENABLE 1 register
m_xfer_done = false;

uint8_t Interrupt_enable1_reg_addr = MAX30105_INTERRUPTENABLE1;
uint8_t Interrutp_enable1_conf_byte = 0xf0;

uint8_t conf_INTEN1[2] = {Interrupt_enable1_reg_addr,
    Interrutp_enable1_conf_byte};

err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address, conf_INTEN1,
    sizeof(conf_INTEN1), false);
printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

while(!nito){
//lettura dato IR LED (se valore >= 1023 oggetto rilevato)
//TX : invio slave address + address del registro FIFO_DATA
m_xfer_done = false;
uint8_t value_FIFO_data = MAX30105_FIFO_DATA;

err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address, &value_FIFO_data,
     1, false);
printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;
//RX: invio slave address + leggo il registro FIFO_DATA (si usano 2 leds
    pertanto si avranno 6 letture per ogni sample)
m_xfer_done = false;
err_code = nrf_drv_twi_rx(&m_twi, pulse_ox_address, &FIFO_data, sizeof
    (FIFO_data));
printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
```

```
while (m_xfer_done == false)
;
if (i == 0){ // devo considerare solo i bits 17 e 16
vettore = FIFO_data << 6;
FIFO_data = vettore >> 6;
IR_data[i] = FIFO_data;
}
else {

if(i > 2){ // stampo a video il sample (3 bytes) dell'ir adc letto
ir_value = (int)IR_data[0] + (int)IR_data[1] + (int)IR_data[2];
printf("valore IR letto : %d [counts] \n\n", ir_value);
nito = true;
}
else
IR_data[i] = FIFO_data;
}

i ++;
bsp_board_led_invert(0);
nrf_delay_ms(200);

}

}

/**
* @brief Function used to congure the MAX30205 temperature sensor
*/
void MAX30205_set_mode() //passo per valore il vettore degli indirizzi dei
     TWI devices trovati dopo lo "scanning"
{
ret_code_t err_code;

printf("Indirizzo sensore temperatura: 0x%x. \n", temp_sensor_address);

/* sensore temperatura: registro CONF : 8 bits D0,D1,.....,D7 */
//conguro un bit alla volta

//conguro bit D0
uint8_t confD0[2] = {MAX30205_REG_CONF, shutdown_modeD0};
```

```
err_code = nrf_drv_twi_tx(&m_twi, temp_sensor_address, confD0, sizeof(
    confD0), false);
// printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

//conguro bit D1
uint8_t confD1[2] = {MAX30205_REG_CONF, OScomparator_modeD1};
m_xfer_done = false;
err_code = nrf_drv_twi_tx(&m_twi, temp_sensor_address, confD1, sizeof(
    confD1), false);
// printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

//conguro bit D2
uint8_t confD2[2] = {MAX30205_REG_CONF, OSpolarityD2};
m_xfer_done = false;
err_code = nrf_drv_twi_tx(&m_twi, temp_sensor_address, confD2, sizeof(
    confD2), false);
// printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

//conguro bit D3
uint8_t confD3[2] = {MAX30205_REG_CONF, D3};
m_xfer_done = false;
err_code = nrf_drv_twi_tx(&m_twi, temp_sensor_address, confD3, sizeof(
    confD3), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

//conguro bit D4
uint8_t confD4[2] = {MAX30205_REG_CONF, D4};
m_xfer_done = false;
err_code = nrf_drv_twi_tx(&m_twi, temp_sensor_address, confD4, sizeof(
```

```
    confD4), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

//conguro bit D5
uint8_t confD5[2] = {MAX30205_REG_CONF, normal_dataD5};
m_xfer_done = false;
err_code = nrf_drv_twi_tx(&m_twi, temp_sensor_address, confD5, sizeof(
    confD5), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

//conguro bit D6
uint8_t confD6[2] = {MAX30205_REG_CONF, bus_timeoutD6};
m_xfer_done = false;
err_code = nrf_drv_twi_tx(&m_twi, temp_sensor_address, confD6, sizeof(
    confD6), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

//conguro bit D7
uint8_t confD7[2] = {MAX30205_REG_CONF, one_shotD7};
m_xfer_done = false;
err_code = nrf_drv_twi_tx(&m_twi, temp_sensor_address, confD7, sizeof(
    confD7), false);
// printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

/* sensore temperatura, registro THYST, setto temperatura minima*/

// m_xfer_done = false;

uint8_t Thyst[2] = {MAX30205_REG_THYST, Tmin};
```

```
m_xfer_done = false;
err_code = nrf_drv_twi_tx(&m_twi, temp_sensor_address, Thyst, sizeof(
    Thyst), false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

/*sensore temperatura, registro TOS, setto temperatura massima*/
uint8_t Tos[2] = {MAX30205_REG_TOS, Tmax};
m_xfer_done = false;
err_code = nrf_drv_twi_tx(&m_twi, temp_sensor_address, Tos, sizeof(Tos),
    false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;
}

/**
* @brief Function used to start a new conversion (new temperature reading)
*/
void start_new_read() {

ret_code_t err_code;

// disabilito il bus  ¯ timeout D6 = '1'
uint8_t confD6[2] = {MAX30205_REG_CONF, 1};
m_xfer_done = false;
err_code = nrf_drv_twi_tx(&m_twi, temp_sensor_address, confD6, sizeof(
    confD6), false);
// printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

// esco dalla shut ¯ down mode D0 = '0'
uint8_t confD0[2] = {MAX30205_REG_CONF, 0};
m_xfer_done = false;
err_code = nrf_drv_twi_tx(&m_twi, temp_sensor_address, confD0, sizeof(
    confD0), false);
```

```
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;
}
```

A.2.5 Menu Utente

```
// comandi per l'utente
void user_display(void)
{
printf("**********Menu Utente********** \n\n\n");
printf("Inserisci il carattere %c per leggere la temperatura con il MAX30105 \n\
    n", 't');
printf("Inserisci il carattere %c per leggere la temperatura con il MAX30205 \n\
    n", 'p');
printf("Inserisci il carattere %c per leggere l'accelerazione angolare con l'
    LSM6DS0 \n\n", 'g');
printf("Inserisci il carattere %c testare funzionamento accelerometro con l'
    LSM6DS0 \n\n", 'l');
printf("Inserisci il carattere %c per leggere l'accelerazione lineare con l'LSM6DS0
     \n\n", 'a');
//printf("Inserisci il carattere %c per leggere l'orientazione spaziale con il
    LIS3MDL \n\n", 'm');
printf("Inserisci il carattere %c per uscire dal programma \n\n", 'e');
}
```

A.2.6 Funzioni acquisizione dati dai sensori

```
/**
* @brief Function for reading data from temperature sensor.
*/
static void read_sensor_data() {
m_xfer_done = false;

/* Leggo 1 byte di dati alla volta, all'indirizzo specicato per il sensore di
    temperatura */
uint8_t value = MAX30205_REG_TEMP;
//1^a fase la TX : invio di slave address + pointer byte da parte del master
    verso il sensore di temperatura (slave)
ret_code_t err_code = nrf_drv_twi_tx(&m_twi, temp_sensor_address, &
    value, 1, false);
```

```
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

m_xfer_done = false;

//2^a fase la RX: nuovo start da parte del master; successivo invio dello slave
    address , e a questo punto si possono leggere i data bytes dal
// registro di temperatura dello slave
err_code = nrf_drv_twi_rx(&m_twi, temp_sensor_address, &m_sample,
    sizeof(m_sample));
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

t = false;
}

/**
* @brief Function used to read temperature measured by the MAX30105 pulse
    oximeter internal temperature sensor
*/
void MAX30105_read_temp() {
ret_code_t err_code;
/*lettura temperatura misurata dal sensore di temp interno al pulse - oximeter
    sensor */
/*leggo prima la parte di TEMP_INT */
m_xfer_done = false;

/* Leggo 1 byte di dati alla volta, all'indirizzo specicato per il sensore pulse -
    oximeter */
uint8_t value_tempint = MAX30105_TEMPINT;
//1^a fase la TX : invio di slave address + pointer byte da parte del master
    verso il sensore di temperatura (slave)
err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address, &value_tempint, 1,
    false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;
```

```
m_xfer_done = false;

//2^a fase la RX: nuovo start da parte del master; successivo invio dello slave
    address , e a questo punto si possono leggere i data bytes dal
// registro di temperatura dello slave
err_code = nrf_drv_twi_rx(&m_twi, pulse_ox_address, &temp_int, sizeof(
    temp_int));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

/*a questo punto leggo la parte di TEMP_FRAC */
m_xfer_done = false;

/* Leggo 1 byte di dati alla volta, all'indirizzo specicato per il sensore pulse
    oximeter */
uint8_t value_tempfrac = MAX30105_TEMPFRAC;
//1^a fase la TX : invio di slave address + pointer byte da parte del master
    verso il sensore di temperatura (slave)
err_code = nrf_drv_twi_tx(&m_twi, pulse_ox_address, &value_tempfrac, 1,
    false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

m_xfer_done = false;

//2^a fase la RX: nuovo start da parte del master; successivo invio dello slave
    address , e a questo punto si possono leggere i data bytes dal
// registro di temperatura dello slave
err_code = nrf_drv_twi_rx(&m_twi, pulse_ox_address, &temp_frac, sizeof(
    temp_frac));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

temp_decimal = calc_temp_frac(temp_frac);
nrf_delay_ms(200);
```

```
printf("Temp misurata con il pulse ⁻ oximeter: %d.%3d Celsius degrees. \n ",
     temp_int , (int)(temp_decimal*PRECISION));
 p = false;
}

int LSM6DS0_read_status_reg17()
{
ret_code_t err_code;
m_xfer_done = false;
uint8_t status_reg_value, XLDA = 0x00, GDA = 0x00 ;
int status;


uint8_t status_reg17_addr = LSM6DS0_STATUS_REG_17;

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &
     status_reg17_addr, 1, false);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;
m_xfer_done = false;

err_code = nrf_drv_twi_rx(&m_twi, accel_gyr_address, &status_reg_value,
     sizeof(status_reg_value));
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

XLDA = status_reg_value << 7;
GDA = status_reg_value << 6;
if (a) // se sto usando l'accelerometro
{
if (XLDA == 0x80)
{
status = 1;
printf("Nuovo sample disponibile dall'accelerometro \n\n");
}
else
{
status = 0;
printf("Nuovo sample non disponibile dall'accelerometro \n\n");
```

```
}
}

if (g) // se sto usando il giroscopio
{
if (GDA == 0xc0)
{
status = 1;
printf("Nuovo sample disponibile dal giroscopio \n\n");
}
else
{
status = 0;
printf("Nuovo sample non disponibile dal giroscopio \n\n");
}

}


return status;
}

void LSM6DS0_read_accelerometer_data()
{
int8_t out_x_l_xl, out_x_h_xl, out_y_l_xl, out_y_h_xl, out_z_l_xl,
      out_z_h_xl; //accelerometer
int16_t vettore_X = 0x0000U, vettore_Y = 0x0000U, vettore_Z = 0x0000U;
ret_code_t err_code;
oat acc_x = 0, acc_y = 0, acc_z = 0;

oset_asseX = 0;
oset_asseY = 0;
oset_asseZ = 0;
//lettura registri dell'accelerometro
m_xfer_done = false;

nrf_delay_ms(200);
// registro OUT_X_L_XL
uint8_t out_x_l_acc_addr = LSM6DS0_OUT_X_L_XL;

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &
```

```
    out_x_l_acc_addr, 1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

m_xfer_done = false;

err_code = nrf_drv_twi_rx(&m_twi, accel_gyr_address, &out_x_l_xl,
    sizeof(out_x_l_xl));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

OUT_X_XL = out_x_l_xl;

//registro OUT_X_H_XL
m_xfer_done = false;

nrf_delay_ms(200);
uint8_t out_x_h_acc_addr = LSM6DS0_OUT_X_H_XL;

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &
    out_x_h_acc_addr, 1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

m_xfer_done = false;

err_code = nrf_drv_twi_rx(&m_twi, accel_gyr_address, &out_x_h_xl,
    sizeof(out_x_h_xl));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

vettore_X = out_x_h_xl;
vettore_X = vettore_X << 8;

OUT_X_XL = OUT_X_XL^vettore_X;

m_xfer_done = false;
```

```
nrf_delay_ms(200);
//registro OUT_Y_L_XL

uint8_t out_y_l_acc_addr = LSM6DS0_OUT_Y_L_XL;

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &
    out_y_l_acc_addr, 1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

m_xfer_done = false;

err_code = nrf_drv_twi_rx(&m_twi, accel_gyr_address, &out_y_l_xl,
    sizeof(out_y_l_xl));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

OUT_Y_XL = out_y_l_xl;

m_xfer_done = false;

nrf_delay_ms(200);
//registro OUT_Y_H_XL

uint8_t out_y_h_acc_addr = LSM6DS0_OUT_Y_H_XL;

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &
    out_y_h_acc_addr, 1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

m_xfer_done = false;

err_code = nrf_drv_twi_rx(&m_twi, accel_gyr_address, &out_y_h_xl,
    sizeof(out_y_h_xl));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
```

```
while (m_xfer_done == false);

vettore_Y = out_y_h_xl;
vettore_Y = vettore_Y << 8;

OUT_Y_XL = OUT_Y_XL^vettore_Y;

m_xfer_done = false;

nrf_delay_ms(200);
//registro OUT_Z_L_XL
uint8_t out_z_l_acc_addr = LSM6DS0_OUT_Z_L_XL;

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &
    out_z_l_acc_addr, 1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

m_xfer_done = false;

err_code = nrf_drv_twi_rx(&m_twi, accel_gyr_address, &out_z_l_xl,
    sizeof(out_z_l_xl));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

OUT_Z_XL = out_z_l_xl;

m_xfer_done = false;

nrf_delay_ms(200);
//registro OUT_Z_H_XL
uint8_t out_z_h_acc_addr = LSM6DS0_OUT_Z_H_XL;


err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &
    out_z_h_acc_addr, 1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);
```

```
m_xfer_done = false;

err_code = nrf_drv_twi_rx(&m_twi, accel_gyr_address, &out_z_h_xl,
      sizeof(out_z_h_xl));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

vettore_Z = out_z_h_xl;
vettore_Z = vettore_Z << 8;

OUT_Z_XL = OUT_Z_XL^vettore_Z;

if (X_axis) // correggo l'oset su asse Y e asse Z
{
oset_asseY = OUT_Y_XL;
oset_asseZ = OUT_Z_XL;
oset_asseX = 0x0000;
X_axis = false;
}
if (Y_axis) // correggo l'oset su asse X e asse Z
{
oset_asseX = OUT_X_XL;
oset_asseZ = OUT_Z_XL;
oset_asseY = 0x0000;
Y_axis = false;
}
if (Z_axis) // correggo l'oset su asse X e asse Y
{
oset_asseX = OUT_X_XL;
oset_asseY = OUT_Y_XL;
oset_asseZ = 0x0000;
Z_axis = false;
}

}

/**
* @brief Function used to read the output data registers of the gyroscope (
      iNemo LSM6DS0)
```

```
*/

void LSM6DS0_read_gyroscope_data(void)
{
int8_t out_x_l_g, out_x_h_g, out_y_l_g, out_y_h_g, out_z_l_g,
      out_z_h_g; // gyroscope
uint16_t vettore = 0x0000U;
ret_code_t err_code;

//lettura registri del giroscopio
m_xfer_done = false;

nrf_delay_ms(200);
//registro OUT_X_L_G
uint8_t outx_l_gyr_addr = LSM6DS0_OUT_X_L_G;

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &outx_l_gyr_addr,
       1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

m_xfer_done = false;

err_code = nrf_drv_twi_rx(&m_twi, accel_gyr_address, &out_x_l_g, sizeof
      (out_x_l_g));
printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

OUT_X_G = out_x_l_g;

//registro OUT_X_H_G
m_xfer_done = false;
nrf_delay_ms(200);
uint8_t outx_h_gyr_addr = LSM6DS0_OUT_X_H_G;

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &outx_h_gyr_addr
      , 1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
```

```
while (m_xfer_done == false);


m_xfer_done = false;

err_code = nrf_drv_twi_rx(&m_twi, accel_gyr_address, &out_x_h_g,
    sizeof(out_x_h_g));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

vettore = out_x_h_g;
vettore = vettore << 8;

OUT_X_G = OUT_X_G^vettore;
nrf_delay_ms(300);
//printf("Accelerazione angolare misurata lungo asse X: %d [dps] \n\n", (int)
    OUT_X_G);
vettore = 0x00U;


//registro OUT_Y_L_G
uint8_t outy_l_gyr_addr = LSM6DS0_OUT_Y_L_G;

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &outy_l_gyr_addr,
     1, false);
// printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

m_xfer_done = false;

nrf_delay_ms(200);

err_code = nrf_drv_twi_rx(&m_twi, accel_gyr_address, &out_y_l_g, sizeof
    (out_y_l_g));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

OUT_Y_G = out_y_l_g;
```

```
//registro OUT_Y_H_G
m_xfer_done = false;
nrf_delay_ms(200);
uint8_t outy_h_gyr_addr = LSM6DS0_OUT_Y_H_G;

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &outy_h_gyr_addr
    , 1, false);
// printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

m_xfer_done = false;
nrf_delay_ms(200);
err_code = nrf_drv_twi_rx(&m_twi, accel_gyr_address, &out_y_h_g,
    sizeof(out_y_h_g));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

vettore = out_y_h_g;
vettore = vettore << 8;

OUT_Y_G = OUT_Y_G^vettore;
nrf_delay_ms(300);
//printf("Accelerazione angolare misurata lungo asse Y: %d [dps] \n\n", (int)
    OUT_Y_G);
vettore = 0x00U;
//nrf_delay_ms(200);

//registro OUT_Z_L_G
m_xfer_done = false;

uint8_t outz_l_gyr_addr = LSM6DS0_OUT_Z_L_G;

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &outz_l_gyr_addr,
     1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);
```

```
m_xfer_done = false;
nrf_delay_ms(200);
err_code = nrf_drv_twi_rx(&m_twi, accel_gyr_address, &out_z_l_g, sizeof
    (out_z_l_g));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

OUT_Z_G = out_z_l_g;

//registro OUT_Z_H_G
m_xfer_done = false;
nrf_delay_ms(200);
uint8_t outz_h_gyr_addr = LSM6DS0_OUT_Z_H_G;

err_code = nrf_drv_twi_tx(&m_twi, accel_gyr_address, &outz_h_gyr_addr
    , 1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

m_xfer_done = false;
nrf_delay_ms(200);
err_code = nrf_drv_twi_rx(&m_twi, accel_gyr_address, &out_z_h_g,
    sizeof(out_z_h_g));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

vettore = out_z_h_g;
vettore = vettore << 8;
OUT_Z_G = OUT_Z_G^vettore;

//nrf_delay_ms(300);
//printf("Accelerazione angolare misurata lungo asse Z: %d [dps] \n\n", (int)
    OUT_Z_G);

nrf_delay_ms(300);
}

void LSM6DS0_correzione_oset(int16_t X_oset, int16_t Y_oset, int16_t
```

```
    Z_oset)
{

//correzione oset lungo asse X
OUT_X_XL = OUT_X_XL      - abs(X_oset);
//correzione oset lungo asse Y
OUT_Y_XL = OUT_Y_XL      - abs(Y_oset);
//correzione oset lungo asse Z
OUT_Z_XL = OUT_Z_XL      - abs(Z_oset);

SEGGER_RTT_printf(0,"Calcolo eventuali oset eseguito correttamente \n\n\
    n");
_Delay(10);
}

int* calcolo_decimal_part(oat a_X, oat a_Y, oat a_Z)
{
static int tri_axis[6];
int a_X_intPart = 0, a_Y_intPart = 0, a_Z_intPart = 0;
int a_X_decimalPart = 0, a_Y_decimalPart = 0, a_Z_decimalPart = 0;

a_X_intPart = (int)a_X;
a_Y_intPart = (int)a_Y;
a_Z_intPart = (int)a_Z;

a_X_decimalPart = abs((a_X   - a_X_intPart)*PRECISION);
a_Y_decimalPart = abs((a_Y   - a_X_intPart)*PRECISION);
a_Z_decimalPart = abs((a_Z   - a_Z_intPart)*PRECISION);


tri_axis[0] = a_X_intPart;
tri_axis[1] = a_X_decimalPart;
tri_axis[2] = a_Y_intPart;
tri_axis[3] = a_Y_decimalPart;
tri_axis[4] = a_Z_intPart;
tri_axis[5] = a_Z_decimalPart;



return tri_axis;
```

```
}

void LIS3MDL_calcolo_orientazione_spaziale(int16_t outm_x, int16_t
     outm_y) //considero solo asse X e asse Y
{
int Xgauss_LSB, Ygauss_LSB,direction_intPart = 0, direction_decPart = 0;
int magn_array[4],i;
oat Xgauss_data = 1, Ygauss_data = 1;
oat angle_rad = 0, angle_deg = 0, direction = 0;


Xgauss_LSB = (int)outm_x;
Ygauss_LSB = (int)outm_y;

// valori dei dati in uscita espressi in [gauss]
Xgauss_data = ((oat)GN_sensitivity/Xgauss_LSB); //
Ygauss_data = ((oat)GN_sensitivity/Ygauss_LSB);

//angolo in [rad] tra la componente X e la componente Y del campo magnetico
angle_rad = atan2f(Ygauss_data, Xgauss_data);
//angolo in [deg]
angle_deg = angle_rad*(180/PI);

// valutazione della direzione
if (Ygauss_data > 0)
direction = 90 − angle_deg;
else
{
if (Ygauss_data < 0)
direction = 270 − angle_deg;
}

if (Ygauss_data == 0)
{
if (Xgauss_data < 0)
direction = 180;
else
{
if (Xgauss_data > 0)
direction = 0;
}
```

```
}
else //in tutti gli altri casi
direction = angle_deg;


direction_intPart = (int)direction;
direction_decPart = (direction  −  direction_intPart)*100;


// determinare se direzione: N, NW, W, SW, S , SE, E, NE
if ((direction > 337.25) || (direction < 22.5)) //North
{
_Delay(20);
SEGGER_RTT_printf(0, "La direzione     %d.%2d gradi N \n",
      direction_intPart, direction_decPart);
_Delay(20);
}
if ((direction > 292.5) && (direction <= 337.25)) //North      − West
{
_Delay(20);
SEGGER_RTT_printf(0, "La direzione     %d.%2d gradi NW \n",
      direction_intPart, direction_decPart);
_Delay(20);
}
if ((direction > 247.5) && (direction <= 292.5)) // West
{
_Delay(20);
SEGGER_RTT_printf(0, "La direzione     %d.%2d gradi W \n",
      direction_intPart, direction_decPart);
_Delay(20);
}
if ((direction > 202.5) && (direction <= 247.5)) // South      − West
{
_Delay(20);
SEGGER_RTT_printf(0, "La direzione     %d.%2d gradi SW \n",
      direction_intPart, direction_decPart);
_Delay(20);
}
if ((direction > 157.5) && (direction <= 202.5)) // South
{
_Delay(20);
```

```
SEGGER_RTT_printf(0, "La direzione     %d.%2d gradi S \n", direction_intPart
    , direction_decPart);
_Delay(20);
}
if ((direction > 112.5) && (direction <= 157.5)) // South      East
{
_Delay(20);
SEGGER_RTT_printf(0, "La direzione     %d.%2d gradi SE \n",
    direction_intPart, direction_decPart);
_Delay(20);
}
if ((direction > 67.5) && (direction <= 112.5)) // East
{
_Delay(20);
SEGGER_RTT_printf(0, "La direzione     %d.%2d gradi E \n",
    direction_intPart, direction_decPart);
_Delay(20);
}
if ((direction > 0) && (direction <= 67.5)) // North       East
{
_Delay(20);
SEGGER_RTT_printf(0, "La direzione     %d.%2d gradi NE \n",
    direction_intPart, direction_decPart);
_Delay(20);
}
if (direction == 0) // Esattamente il polo Nord geograco
{
_Delay(20);
SEGGER_RTT_printf(0, "La bussola indica esattamente il polo Nord
    geograco \n");
_Delay(20);
}
}

void LSM6DS0_calcolo_accelerazione_treassi(int16_t out_x, int16_t out_y,
    int16_t out_z)
{
uint16_t resultX = 0, resultY = 0 , resultZ = 0;
uint16_t mask = 0x0000;
int N_bit = 16,i;
int MSB_resultX, MSB_resultY, MSB_resultZ;
```

```
int asseX_LSB, asseY_LSB, asseZ_LSB;
int MSB_X, MSB_Y, MSB_Z;
int acc_array[6];
oat accel_X_dps = 1, accel_Y_dps = 1, accel_Z_dps = 1;
oat accel_X_g = 1, accel_Y_g = 1, accel_Z_g = 1;
bool pos_X = false, pos_Y = false, pos_Z = false;
bool neg_X = false, neg_Y = false, neg_Z = false;

//devo convertire le uscite dell'accelerometro considerando se queste sono
    positive o negative
resultX = out_x^mask;
resultY = out_y^mask;
resultZ = out_z^mask;

//valuto i MSBs delle "outs" e i MSBs dei "results"
MSB_X = out_x >> (N_bit      1);
MSB_Y = out_y >> (N_bit      1);
MSB_Z = out_z >> (N_bit      1);

MSB_resultX = resultX >> (N_bit      1);
MSB_resultY = resultY >> (N_bit      1);
MSB_resultZ = resultZ >> (N_bit      1);

if(MSB_X == MSB_resultX)
pos_X = true;
else
neg_X = true;

if(MSB_Y == MSB_resultY)
pos_Y =true;
else
neg_Y = true;

if(MSB_Z == MSB_resultZ)
pos_Z = true;
else
neg_Z = true;

if (a) // se sto usando l'accelerometro
{
asseX_LSB = (int)out_x;
```

```
asseY_LSB = (int)out_y;
asseZ_LSB = (int)out_z;

// converto inne per ottenere l'accelerazione espressa in [g] e non in LSB
accel_X_g = asseX_LSB*LA_sensitivity;
accel_Y_g = asseY_LSB*LA_sensitivity;
accel_Z_g = asseZ_LSB*LA_sensitivity;

for (i = 0; i<6; i++ )
acc_array[i] = calcolo_decimal_part(accel_X_g, accel_Y_g, accel_Z_g)[i];

if (pos_X)
SEGGER_RTT_printf(0,"Accelerazione lineare lungo l'asse X : %c %d.%3d [g]
    \n\n ", '+', acc_array[0], acc_array[1]);
if (pos_Y)
SEGGER_RTT_printf(0,"Accelerazione lineare lungo l'asse Y : %c %d.%3d [g]
    \n\n ", '+', acc_array[2], acc_array[3]);
if (pos_Z)
SEGGER_RTT_printf(0,"Accelerazione lineare lungo l'asse Z : %c %d.%3d [g]
    \n\n ", '+', acc_array[4], acc_array[5]);


if (neg_X)
SEGGER_RTT_printf(0,"Accelerazione lineare lungo l'asse X : %d.%3d [g] \n\
    n ", acc_array[0], acc_array[1]);
if (neg_Y)
SEGGER_RTT_printf(0,"Accelerazione lineare lungo l'asse Y : %d.%3d [g] \n\
    n ", acc_array[2], acc_array[3]);
if (neg_Z)
SEGGER_RTT_printf(0,"Accelerazione lineare lungo l'asse Z : %d.%3d [g] \n\
    n ", acc_array[4], acc_array[5]);


}

if (g) // se sto usando il giroscopio
{
asseX_LSB = (int)out_x;
asseY_LSB = (int)out_y;
asseZ_LSB = (int)out_z;
```

```
// converto inne per ottenere l'accelerazione angolare espressa in [dps] e non in
    LSB
accel_X_dps = asseX_LSB*G_sensitivity;
accel_Y_dps = asseY_LSB*G_sensitivity;
accel_Z_dps = asseZ_LSB*G_sensitivity;

for (i = 0; i<6; i++ )
acc_array[i] = calcolo_decimal_part(accel_X_dps, accel_Y_dps, accel_Z_dps
    )[i];


if (pos_X)
SEGGER_RTT_printf(0,"Accelerazione angolare lungo l'asse X : %c %d.%3d [
    dps] \n\n ", '+', acc_array[0], acc_array[1]);
if (pos_Y)
SEGGER_RTT_printf(0,"Accelerazione angolare lungo l'asse Y : %c %d.%3d [
    dps] \n\n ", '+', acc_array[2], acc_array[3]);
if (pos_Z)
SEGGER_RTT_printf(0,"Accelerazione angolare lungo l'asse Z : %c %d.%3d [
    dps] \n\n ", '+', acc_array[4], acc_array[5]);


if (neg_X)
SEGGER_RTT_printf(0,"Accelerazione angolare lungo l'asse X : %d.%3d [dps]
    \n\n ", acc_array[0], acc_array[1]);
if (neg_Y)
SEGGER_RTT_printf(0,"Accelerazione angolare lungo l'asse Y : %d.%3d [dps]
    \n\n ", acc_array[2], acc_array[3]);
if (neg_Z)
SEGGER_RTT_printf(0,"Accelerazione angolare lungo l'asse Z : %d.%3d [dps]
    \n\n ", acc_array[4], acc_array[5]);

}

}


/**
* @brief Function used to read the output data registers of the magnetometer (
    LIS3MDL)
*/
```

```
void LIS3MDL_read_magnetometer_data()
{
int8_t out_x_l, out_x_h, out_y_l, out_y_h, out_z_l, out_z_h; //
    magnetometer
int16_t vettore_X = 0x0000U, vettore_Y = 0x0000U, vettore_Z = 0x0000U;
ret_code_t err_code;

//lettura registri d'uscita del magnetometro
m_xfer_done = false;

nrf_delay_ms(200);
//registro OUT_X_L
uint8_t out_x_l_magn_addr = LIS3MDL_OUT_X_L;

err_code = nrf_drv_twi_tx(&m_twi, magnetometer_address, &
    out_x_l_magn_addr, 1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

m_xfer_done = false;

err_code = nrf_drv_twi_rx(&m_twi, magnetometer_address, &out_x_l,
    sizeof(out_x_l));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

OUT_X = out_x_l;

//registro OUT_X_H
m_xfer_done = false;

nrf_delay_ms(200);
uint8_t out_x_h_magn_addr = LIS3MDL_OUT_X_H;

err_code = nrf_drv_twi_tx(&m_twi, magnetometer_address, &
    out_x_h_magn_addr, 1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);
```

```
m_xfer_done = false;

err_code = nrf_drv_twi_rx(&m_twi, magnetometer_address, &out_x_h,
    sizeof(out_x_h));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

vettore_X = out_x_h;
vettore_X = vettore_X << 8;

OUT_X = OUT_X^vettore_X;

//registro OUT_Y_L
m_xfer_done = false;

nrf_delay_ms(200);
uint8_t out_y_l_magn_addr = LIS3MDL_OUT_Y_L;

err_code = nrf_drv_twi_tx(&m_twi, magnetometer_address, &
    out_y_l_magn_addr, 1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

m_xfer_done = false;

err_code = nrf_drv_twi_rx(&m_twi, magnetometer_address, &out_y_l,
    sizeof(out_y_l));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

OUT_Y = out_y_l;

//registro OUT_Y_H
m_xfer_done = false;

nrf_delay_ms(200);
uint8_t out_y_h_magn_addr = LIS3MDL_OUT_Y_H;
```

```
err_code = nrf_drv_twi_tx(&m_twi, magnetometer_address, &
    out_y_h_magn_addr, 1, false);
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

m_xfer_done = false;

err_code = nrf_drv_twi_rx(&m_twi, magnetometer_address, &out_y_h,
    sizeof(out_y_h));
//printf("Error code : %d \n", err_code);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false);

vettore_Y = out_y_h;
vettore_Y = vettore_Y << 8;

OUT_Y = OUT_Y^vettore_Y;

}

int LIS3MDL_read_status_reg()
{
ret_code_t err_code;
m_xfer_done = false;
uint8_t stat_reg_value, XDA = 0x00, YDA = 0x00;
int statusX_m = 0, statusY_m = 0;
int stati_m[2];

uint8_t stat_reg_addr = LIS3MDL_STATUS_REG;

nrf_delay_ms(200);
//leggo il bit XDA
err_code = nrf_drv_twi_tx(&m_twi, magnetometer_address, &
    stat_reg_addr, 1, false);
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;
m_xfer_done = false;
```

```
err_code = nrf_drv_twi_rx(&m_twi, magnetometer_address, &
     stat_reg_value, sizeof(stat_reg_value));
APP_ERROR_CHECK(err_code);
while (m_xfer_done == false)
;

XDA = stat_reg_value << 7;
YDA = stat_reg_value << 6;

if (XDA == 0x80){
printf("Nuovo sample disponibile lungo asse X \n");
statusX_m = 1;
}
else
{
printf("Nuovo sample non disponibile lungo asse X");
statusX_m = 0;
}

if (YDA == 0xC0){
printf("Nuovo sample disponibile lungo asse Y \n");
statusY_m = 1;
}
else
{
printf("Nuovo sample non disponibile lungo asse Y \n");
statusY_m = 0;
}

stati_m[0] = statusX_m;
stati_m[1] = statusY_m;

//return stati_m;

}
```

Appendice B

Protocollo di comunicazione I2C

Il protocollo di comunicazione seriale usato per la trasmissione/ricezione dati tra il microcontrollore e i sensori digitali, è l'I2C. Si tratta, di un protocollo a due li SDA e SCL, rispettivamente una linea dati bidirezionale e una linea di clock unidirezionale (infatti, il segnale di CLK viene fornito dal MASTER per sincronizzare la comunicazione dei dati con lo SLAVE). Con questo sistema, possono essere messi in comunicazione un MASTER, con numerosi SLAVEs. Tuttavia, con questo protocollo, a dierenza dell'SPI, non è il MASTER che abilita lo slave scelto, bensì sono gli slaves che provocano una variazione del valore logico sulla linea SDA per far capire al Master che sono pronti ad iniziare una comunicazione.Gli slaves, basati sul protocollo I2C, sono caratterizzati, infatti, da un'uscita di tipo open drain, come mostrato nella seguente gura:

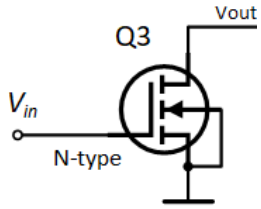


Figura B.1: Uscita open drain, [114]

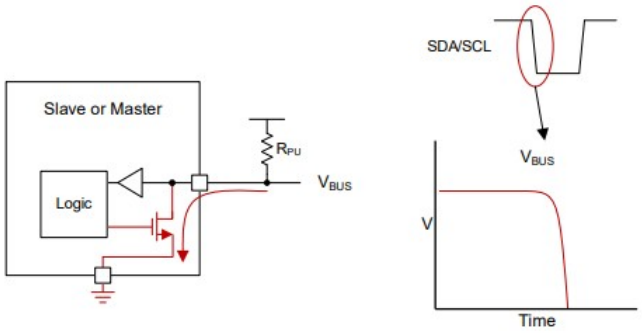


Figura B.2: Open Drain "tira giù" la tensione sulla linea, [115]

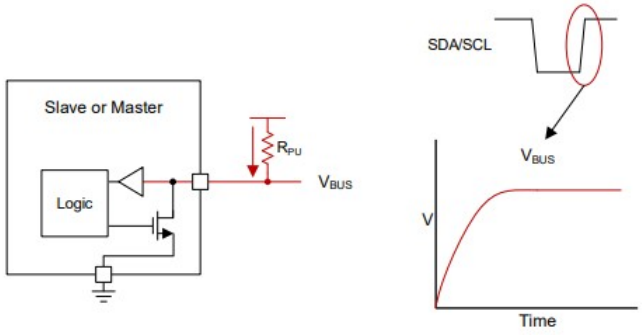


Figura B.3: Open Drain "rilascia" la tensione sulla linea, [115]

In particolare, i dispositivi con uscita di tipo open drain possono pilotare solo il livello LOW ('0') sulla linea SDA; quando uno slave pilota la linea SDA a '0' signica che la comunicazione tra quello slave e il master è iniziata.Tuttavia, per riportare la linea SDA al livello HIGH ('1') è necessario introdurre delle resistenze di pull-up, tra Vout, B.1 (e dunque SDA) e Vdd, ed anche tra SCL e Vdd.In questo modo, è garantita la comunicazione del master con un solo sensore per volta, eliminando di conseguenza gli errori dovuti a comunicazioni simultanee, di due o più sensori.
La comunicazione I2C inizia con una condizione di START e termina con una condizione di STOP. Lo START bit è caratterizzato da una variazione da '1' a '0', o da alto a basso, della linea SDA mentre la linea SCL è al livello logico HIGH ('1'). Al contrario, lo STOP bit è caratterizzato da una variazione da '0' a '1', o da basso af alto, della linea SDA mentre la linea SCL è al livello logico HIGH ('1'). Tali condizioni sono riportate nella seguente gura:

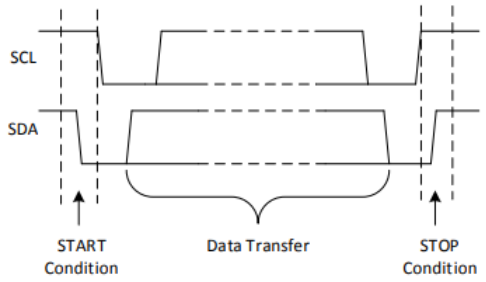


Figura B.4: Condizioni di START e di STOP, [115]

Pertanto, lo START indica il momento in cui il Master inizia la comuncazion seriale portando la linea dati SDA al livello logico bassoInvece, lo STOP indica il momento in cui il Master termina la comunicazione con lo Slave e "rilascia" la linea dati SDA ad un valore logico alto, in modo tale da permettere che altri slaves si possano connettere al BUS di comunicazione.
Per comprendere meglio lo standard I2C, sono state considerati i due tipi di comunicazione principali tra master e slaveE' stato considerato il sensore accelerometro/giroscopio LSM6DS0 come esempio, [79].
Occorre specicare, innanzitutto, che solitamente questi sensori presentano un'indirizzo su 7 bits, poi completato da un'ottavo bit R/W che indica se verrà eseguita un'operazione di scrittura dati dal Master verso lo Slave, o un'operazione di lettura dati dallo Slave verso il Master. Infatti, queste sono le due operazioni principali.

Table 16. Transfer when master is writing one byte to slave

Master	ST	SAD + W		SUB		DATA		SP
Slave			SAK		SAK		SAK	

Figura B.5: Scrittura di un byte da Master verso Slave, [79]

Dalla gura B.5, è facile comprendere che:si ha l'invio del bit di START (ST), dopodichè il Master "seleziona" lo slave con cui vuole comunicare inviando gli 8 bits d'indirizzo dello slave, ossia i 7 bits di SAD (slave address) più il bit W, che in questo caso indica l'operazione di scrittura; quindi lo Slave invia un bit di Acknoledge (SAK), portando la linea SDA a '0' facendo capire al Master che è pronto ad iniziare la comunicazione, B.2; a questo punto il Master invia un sub-address (SUB) ossia l'indirizzo del registro interno al sensore che dev'essere congurato, questo indirizzo sarà seguito da un'altro SAK da parte dello slave; inne, il Master invia gli 8 bits di dato DATA che devono essere scritti nel registro specicato dall'indirizzo SUB. La scrittura termina con uno SAK da parte dello slave seguito dallo STOP bit inviato dal Master.

Table 18. Transfer when master is receiving (reading) one byte of data from slave

Master	ST	SAD + W		SUB		SR	SAD + R			NMAK	SP
Slave			SAK		SAK			SAK	DATA		

Figura B.6: Lettura di un Byte da Slave verso il Master, [79]

La gura B.6 mostra, invece, l'operazione di lettura di 1 Byte da Slave verso il Master. L'operazione di lettura è più complicata, poichè prevede due steps per poterla completare. Il primo step consiste in una scrittura (SAD + W) dello slave address e del sub-address (SUB), che indicano rispettivamente lo slave con cui si vuole comunicare e il registro di dati d'uscita che si vuole leggere. Dopodichè, occorre una seconda condizione di START (SR) per iniziare l'operazione di lettura vera e propria. Pertanto è inviato nuovamente lo slave address ma indicando la lettura (SAD + R), quindi, dopo l'acknoledgment dello slave, è letto il dato di 8 bits dal registro selezionato (DATA). La comunicazione è poi terminata con un acknoledgment da parte del Master (NMAK), che porta la linea ad un valore logico alto, B.3, seguito dalla condizione di STOP.
Inne, è importante sottolineare che con il protocollo I2C, possono essere letti o scritti soltanto 8 bits per volta; quindi, per la lettura e la scrittura di dati su un numero di bits maggiore saranno necessarie operazioni di scrittura e lettura multiple.

Bibliograa

[1] L. Dipietro, A. M. Sabatini, S. Member, and P. Dario, A Survey of Glove-Based Systems and Their Applications, vol. 38, no. 4, pp. 461482, 2008.

[2] Journal of Rehabilitation Research & Development, vol. 40, No. 2, March/April 2003, pp. 179-190

[3] Zimmerman, Lanier et al."A Hand Gesture Interface Device" (1987).

[4] Humanware S.R.L.. Humanglove developers manual; Pisa, Italy; 1998

[5] Williams NW, Penrose JMT, Caddy CM, Barnes E, Hose DR, Harley P. A goniometric glove for clinical hand assessment. J Hand Surg [Br] 2000;25B(2):200-7.

[6] Williams NW. The virtual hand, The Pulvertaft Prize Essay for 1996. J Hand Surg [Br] 1997;22B(5):561-67.

[7] Simone LK, Kamper DG. Design considerations for a wearable monitor to measure nger posture. J Neuroeng Rehabil. 2005;2(5):110.

[8] Simone LK, Sundarrajan N, Luo X, Jia Y, Kamper DG. A low cost instrumented glove for extended monitoring and functional hand assessment. J Neurosci Meth. 2007;160:335348. doi:10.1016/j.jneumeth.2006.09.021.

[9] Lee Swallow, Elias Siores "Tremor Suppression Using Smart Textile Fibre Systems", Journal of Fiber Bioengineering and Informatics; 2009

[10] "Development and evaluation of a low-cost sensor glove for assessment of human nger movements in neurophysiological settings", Journal of Neuroscience Methods, Volume 178, Issue 1, 30 March 2009, Pages 138-147

[11] La Viola JJ. A survey of hand posture and gesture recognition techniques and technology. 1999, CS-99-11, Brown University, Department of Computer Science, Providence (RI).

[12] N. P. Oess, J. Wanek, and A. Curt, "Design and evaluation of a lowcost instrumented glove for hand function assessment," J Neuroeng Rehabil, vol. 9 : 2, 2012.

[13] Aswad A.R, Khairunizam Wan, Nazrul H. ADNAN, Shahriman A.B, D. Hazry and Zuradzman M. Razlan, Nabilah H.E and M.Hazwan ALI "Glove Based Virtual Reality (VR) Interaction for the Purpose of Rehabilitation", Australian Journal of Basic and Applied Sciences, 8(4) Special 2014, Pages: 170-175

[14] http://www.inition.co.uk

[15] M Sivak, D Murray, L Dick, C Mavroidis, M K Holden, "Development of a low-cost virtual reality-based smart glove for rehabilitation", Proc. 9th Intl Conf. Disability, Virtual Reality & Associated Technologies Laval, France, 1012 Sept. 2012

[16] Brendan O'Flynn, Javier Torres Sanchez, Philip Angove, James Connolly, Joan Condell, Kevin Curran, Philip Gardiner, "Novel Smart Sensor Glove for Arthritis Rehabilitation", Conference: Body Sensor Networks (BSN), May 2013 IEEE International Conference on

[17] Gaurav Singh, Sravya Boddu, Indronee Chakravorty, G. Muralidhar Bairy, and Ganesh M., "An Instrumented Glove for Monitoring Forces During Object Manipulation", 2013 IEEE Point-of-Care Healthcare Technologies (PHT) Bangalore, India, 16 - 18 January, 2013

[18] https://pressureprofile.com/fingertps

[19] Zhojie Ju, Honghai Liu, "Human Hand Motion Analysis with Multisensory Information", IEEE/ASME TRANSACTIONS ON MECHATRONICS, VOL.19, NO.2, APRIL 2014.

[20] Gereon H.Buscher, Risto Koiva, Carsten Schurmann, Robert Haschke, Helge J.Ritter, "Flexible and stretchable fabric-based tactile sensor", Robotics and Autonomous Systems, Volume 63, Part 3, January 2015, Pages 244-252.

[21] Rafael Tavares, Paulo Abreu, Manuel Quintas, "Instrumented Glove for Rehabilitation Exercises"

[22] Rafael Tavares and Pedro José Sousa, Paulo Abreu and Maria Teresa Restivo, "Virtual environment for instrumented glove"

[23] Yang Zheng, Yu Peng, Gang Wang, Xinrong Liu, Xiaotong Dong, Jue Wanga, "Development and evaluation of a sensor glove for hand function assessment and preliminary attempts at assessing hand coordination", Measurement, Volume 93, November 2016, pages 1-12.

[24] Mostafa Mohammadi, Tommaso Lisini Baldi, Stefano Scheggi, and Domenico Prattichizzo, "Fingertip Force Estimation via Inertial and Magnetic Sensors in Deformable Object Manipulation", HAPTICS SYMPOSIUM '16, APRIL 811, 2016, PHILADELPHIA, PENNSYLVANIA, USA

[25] http://www.bioservo.com/healthcare/product-portfolio/sem-glove/

[26] Thompson, W. (2015). Worldwide survey of tness trends for 2016:10th Anniversary Edition. ACSM Health Fitness J, 19, 9-18.

[27] Thompson, W. (2016). Worldwide Survey Of Fitness Trends For 2017. ACSM Health Fitness J, 20(6), 8-17.

[28] Düking, P., Holmberg, H. C., & Sperlich, B. (2017). Instant Biofeedback Provided by Wearable Sensor Technology Can Help to Optimize Exercise and Prevent Injury and Overuse. Front Physiol, 8, 167.

[29] https://www.wearable-technologies.com/2017/09/ wearables-for-rehabilitation-2/

[30] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4168435/

[31] D. J. Sturman and D. Zeltzer, A survey of glove-based input, IEEE Comput. Graph. Appl., vol. 14, no. 1, pp. 3039, Jan. 1994.

[32] D. J. Sturman and D. Zeltzer, A survey of glove-based input, IEEE Comput. Graph. Appl., vol. 14, no. 1, pp. 3039, Jan. 1994.

[33] Laura Dipietro, Angelo M. Sabatini, Paolo Dario, "A Survey of Glove-Based Systems and Their Applications", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICSPART C: APPLICATIONS AND REVIEWS, VOL. 38, NO. 4, JULY 2008

[34] https://www.microsoft.com/buxtoncollection/a/pdf/cg%20a%20glove% 20survey.pdf

[35] http://www.flexpoint.com/

[36] https://it.emcelettronica.com/la-tecnologia-ad-effetto-hall

[37] http://www.5dt.com/data-gloves/

[38] P. Coiet and G. C. Burdea, Virtual Reality Technology. New York: Wiley, 2003.

[39] https://www.interlinkelectronics.com/

[40] https: //www.tekscan.com/flexiforce-load-force-sensors-and-systems

[41] https://eeonyx.com/

[42] https://eeonyx.com/products/ntex-stretchy-sensor/

[43] King H.Yang, "Basic Finite Element Method as Applied to Injury Biomechanics", Chapter 4 Element Stiness Matrix, 2018, Pages 151230, Wayne State University, Detroit, Michigan, United States

[44] https://www.invensense.com/technology/motion/

[45] https://www.cl.cam.ac.uk/~rmf25/papers/Understanding%20the% 20Basis%20of%20the%20Kalman%20Filter.pdf

[46] https://it.mathworks.com/videos/ understanding-kalman-filters-part-1-why-use-kalman-filters--1485813028675. html

[47] David A. Forsyth, Jean Ponce, "Computer Vision: A Modern Approach", Prentice Hall Professional Technical Reference ©2002

[48] https://it.wikipedia.org/wiki/Programma_Apollo

[49] Henk G Kortier, Victor I Sluiter, Daniel Roetenberg, and Peter H Veltink, "Assessment of hand kinematics using inertial and magnetic sensors", Journal of NeuroEngineering and Rehabilitation,2014.

[50] http://www-jlc.kek.jp/2003oct/subg/o/kaltest/doc/ReferenceManual.pdf

[51] https://home.wlu.edu/~levys/kalman_tutorial/

[52] http://www.st.com/en/mems-and-sensors/lsm330dlc.html

[53] https://aerocontent.honeywell.com/aero/common/documents/ myaerospacecatalog-documents/Defense_Brochures-documents/HMC5983_ 3_Axis_Compass_IC.pdf

[54] R L Hadimani, D Vatansever Bayramol, N Sion, T Shah, Limin Qian, Shaoxin Shi and E Siores, "Continuous production of piezoelectric PVDF bre for e-textile applications", Smart Materials and Structures, Volume 22, Number 7, Published 6 June 2013 ® 2013 IOP Publishing Ltd

[55] Giorgio De Pasquale, Sang-Gook Kim, Daniele De Pasquale, "GoldFinger: Wireless human-machine interface with dedicated software and biomechanical energy harvesting system", IEEE/ASME Trans. Mechatronics, vol. 21, no. 1, p. 1, 2015.

[56] L. M. Swallow, J. K. Luo, E. Siores, I. Patel, and D. Dodds, A piezoelectric bre composite based energy harvesting device for potential wearable applications, Smart Mater. Struct., vol. 17, no. FEBRUARY, p. 25017, 2008.

[57] http://www.dataphysics.com/downloads/technical/Understanding-the-Physics-of-Electrodynamic-Shaker-Performance-by-G.F.-Lang-and-D.-Snyder.pdf

[58] M. A. Peres, R. W. Bono, D. L. Brown, "Practical Aspects of Shaker Measurements for Modal Testing ", Cincinnati, Ohio, 45221, USA

[59] N. P. Oess, J. Wanek, and H. J. A. Van Hedel, Enhancement of bend sensor properties as applied in a glove for use in neurorehabilitation settings, 2010 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC'10, pp. 59035906, 2010.

[60] http://www.spectrasymbol.com/product/flex-sensors/

[61] https://www.healthline.com/health/ulnar-deviation

[62] http://www.diee.unica.it/misure/Dispense/Misure_Elettriche_Elettroniche/Sistemi_di_acquisizione_dati.pdf

[63] http://www.analog.com/media/en/technical-documentation/data-sheets/AD7490.pdf

[64] http://www.analog.com/media/en/technical-documentation/data-sheets/AD7888.pdf

[65] http://tesi.cab.unipd.it/26398/1/tesi.pdf

[66] http://home.hit.no/~finnh/videos/topics/labview_analog_io/371303.pdf

[67] http://pdf.datasheetcatalog.com/datasheet/nationalsemiconductor/DS005672.PDF

[68] https://www.nxp.com/docs/en/data-sheet/MC68HC912D60A.pdf

[69] https://www.microchip.com/design-centers/microcontroller

[70] http://ww1.microchip.com/downloads/en/DeviceDoc/
Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.
pdf

[71] http://www.ti.com/lit/ds/symlink/msp430g2453.pdf

[72] http://www.keil.com/dd/docs/datashts/philips/p89v51rd2.pdf

[73] https://www.mouser.com/ds/2/268/doc8077-1066117.pdf

[74] http://ww1.microchip.com/downloads/en/DeviceDoc/
Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_
datasheet.pdf

[75] G. Saggio, S. Bocchetti, C. A. Pinto, and G. Orengo, Wireless data glove
system developed for HMI, 2010 3rd Int. Symp. Appl. Sci. Biomed. Commun.
Technol. ISABEL 2010, pp. 04, 2010.

[76] https://it.emcelettronica.com/
implementare-il-gsr-galvanic-skin-response-a-microcontrollore-nei-dispositivi-indossabili

[77] Leonardo Mastrototaro, ""

[78] https://download.mikroe.com/documents/datasheets/
HPL402323-2C-190mAh.pdf

[79] http://www.st.com/content/ccc/resource/technical/document/
datasheet/6e/09/28/0b/01/06/42/24/DM00101533.pdf/files/
DM00101533.pdf/jcr:content/translations/en.DM00101533.pdf

[80] http://www.st.com/resource/en/datasheet/lis3mdl.pdf

[81] https://datasheets.maximintegrated.com/en/ds/MAX30205.pdf

[82] http://wiki.seeedstudio.com/Grove-GSR_Sensor/

[83] http://www.centropiaggio.unipi.it/sites/default/files/course/
material/Sensori%20fisici_3.pdf

[84] http://www.geostudiastier.it/writable/public/userfiles/I_
magnetometri_scalari_WEB(1).pdf

[85] Giuseppe Tarabella, Marco Villani, Davide Calestani, Roberto Mosca, Salvatore Iannotta, Andrea Zappettini e Nicola Coppedè, "A single cotton ber organic electrochemical transistor for liquid electrolyte saline sensing", Journal of Chemistry, Issue 45, 2012"

[86] Maria Viqueira Villarejo, Begona Garcia Zapirain e Amaia Mèndez Zorrilla, "A Stress Sensor Based on Galvanic Skin Response (GSR) controlled by ZigBee, Sensors, MDPI, Basel 2012"

[87] https://www.seeedstudio.com/Grove-GSR-sensor-p-1614.html

[88] Satu Rajala, Jukka Lekkala, "PVDF and EMFi sensor materials-A comparative study", Proc.Eurosensors XXIV, September 5-8,2010,Linz,Austria

[89] Claire M.Lochner, Yasser Khan, Adrien Pierre, Ana Claudia Arias, "System Design for Organic Pulse Oximeter"

[90] https://www.mouser.com/ds/2/256/MAX30105-967526.pdf

[91] http://www.analog.com/media/en/technical-documentation/ data-sheets/ADG3304.pdf

[92] http://pdf1.alldatasheet.com/datasheet-pdf/view/125210/ETC1/ MJTP1230.html

[93] http://www.ti.com/lit/ds/symlink/tps65721.pdf

[94] https://download.mikroe.com/documents/datasheets/ HPL402323-2C-190mAh.pdf

[95] http://ww1.microchip.com/downloads/en/DeviceDoc/21733j.pdf

[96] https://www.mouser.com/catalog/specsheets/Seeed_101020052.pdf

[97] http://www.ti.com/lit/ds/snosc16d/snosc16d.pdf

[98] https://www.phidgets.com/productfiles/3100/3100_0/Documentation/ 3100_0_FlexiforceUserManual.pdf

[99] https: //www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF52832

[100] https://www.nordicsemi.com/eng/Products/S132-SoftDevice

[101] http://aconno.de/acn52832/

[102] https://www.yuden.co.jp/or/solutions/ble/

[103] https://www.yuden.co.jp/wireless_module/document/datareport2/en/TY_BLE_EYSHSNZWZ_DataReport_V1_0_20170630E.pdf

[104] http://www.interfacebus.com/MicroSD_Card_Pinout.html

[105] http://ww1.microchip.com/downloads/en/DeviceDoc/21189T.pdf

[106] https://assets.nexperia.com/documents/data-sheet/74HC_HCT4051.pdf

[107] https://www.digikey.com/product-detail/en/harwin-inc/P70-7000045/952-3126-ND/6211188

[108] http://support.seeedstudio.com/knowledgebase/articles/447362-fusion-pcb-specification

[109] http://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.nrf52%2Fdita%2Fnrf52%2Fdevelopment%2Fnrf52_dev_kit.html

[110] http://infocenter.nordicsemi.com/pdf/nRF52_DK_User_Guide_v1.2.pdf

[111] http://infocenter.nordicsemi.com/pdf/nRF52_DK_User_Guide_v1.2.pdf

[112] https://www.segger.com/downloads/embedded-studio/

[113] http://www.ti.com/lit/an/slva139/slva139.pdf

[114] https://os.mbed.com/teams/TVZ-Mechatronics-Team/wiki/Digital-inputs-and-outputs

[115] http://www.ti.com/lit/an/slva704/slva704.pdf

[116] https://docs-emea.rs-online.com/webdocs/0fcd/0900766b80fcdd7a.pdf

[117] https://docs-emea.rs-online.com/webdocs/110a/0900766b8110a330.pdf

[118] https://www.amazon.it/Sostegno-riparazioni-allo-smartphone-PCB/dp/B00I6I37EY/ref=pd_sim_107_1?_encoding=UTF8&pd_rd_i=B00I6I37EY&pd_rd_r=76eea449-7922-11e8-b32f-999c3ee1f438&pd_rd_w=TZZZg&pd_rd_wg=aWsOJ&pf_rd_i=desktop-dp-sims&pf_rd_m=A11IL2PNWYJU7H&pf_rd_p=6946954921869226644&pf_rd_r=G02DYYM8JRX2D2XWP0Q2&pf_rd_s=desktop-dp-sims&pf_rd_t=40701&psc=1&refRID=G02DYYM8JRX2D2XWP0Q2

[119] https://it.rs-online.com/web/p/stazioni-di-saldatura/1362587/

[120] https://it.rs-online.com/web/p/dissaldatori-a-pinzetta/7300726/?relevancy-data=636F3D3126696E3D4931384E4C446573635461786F6E6F6D794272616E6453656172
394E267374613D77656C6C657220543030353133331373839394E26

[121] http://whatsyourtech.ca/2015/12/29/review-athos-wearable-tech-shirt-and-shorts

[122] https://www.cnet.com/news/peregrine-glove-for-the-touchy-feely-gamer