



**SAPIENZA**  
UNIVERSITÀ DI ROMA

# Sensor-equipped Eyeglasses

**Facoltà di Ingegneria dell'informazione, informatica e statistica**  
**Corso di laurea in Informatica**

**Valerio Caciolo**  
**Matricola 1923344**

Relatore  
Prof.ssa Maria De Marsico

A.A. 2020-2021

## Sommario

Il seguente elaborato affronta la progettazione e la realizzazione dei “Sensor-equipped Eyeglasses”, degli occhiali connessi e dotati di molteplici sensori, pensati soprattutto, ma non solo, per la fascia anziana della popolazione. Il repository del progetto con codici, immagini ed allegati è presente su GitHub al link:

[https://github.com/ValerioCac/SE\\_Eyeglasses.git](https://github.com/ValerioCac/SE_Eyeglasses.git)

Nei primi capitoli l’elaborato affronta lo studio e la raccolta di informazioni inerenti allo stato di salute di una persona anziana media, definendo i maggiori rischi correlati all’età, quali quelli derivanti da una caduta accidentale e quelli da problemi cardiocircolatori, evidenziando a tal riguardo i benefici di un continuo monitoraggio del soggetto, tramite il quale operare un pronto intervento in caso di malori.

Segue la definizione delle funzionalità di cui devono essere dotati gli occhiali, per poi passare alla scelta dell’hardware, alla produzione tramite stampa 3D ed infine all’assemblaggio di un prototipo funzionante. Nell’ultima sezione dell’elaborato, concentrata sullo sviluppo software, vengono esaminati gli algoritmi utilizzati per il corretto funzionamento dei sensori e i dati registrati. Questi ultimi vengono poi analizzati e confrontati con quelli prodotti da dispositivi simili a quello presentato per permettere l’apporto di varie migliorie agli algoritmi. Viene inoltre mostrato l’utilizzo di Arduino IoT Cloud che, attraverso un software per dispositivi *mobile*, permette la gestione tramite smartphone dei dati raccolti dagli occhiali.

In conclusione, vengono illustrate le aspettative di un prodotto finito, evidenziando tutte le possibili applicazioni future.

# Indice

<b>Cap. 1 - <a href="#">Introduzione</a></b>	4
<b>Cap. 2 - <a href="#">Studi, ricerche e definizione dell'approccio</a></b>	5
a. <a href="#">Internet of Medical Things (IoMT)</a>	5
b. <a href="#">Frequenza cardiaca ed ECG</a>	6
c. <a href="#">Rischio da caduta</a>	7
d. <a href="#">Importanza del pronto intervento</a>	7
e. <a href="#">Come poter monitorare i dati</a>	7
<b>Cap. 3 - <a href="#">Funzionalità</a></b>	8
a. <a href="#">Analisi del contesto</a>	8
b. <a href="#">Requisiti</a>	9
i. <a href="#">Funzionali</a>	9
ii. <a href="#">Non Funzionali</a>	9
c. <a href="#">Il prototipo</a>	10
<b>Cap. 4 - <a href="#">Progettazione Hardware</a></b>	11
a. <a href="#">Scelta dell'hardware</a>	11
i. <a href="#">Emi-occhiale destro: Arduino e sensori</a>	11
ii. <a href="#">Emi-occhiale sinistro: microcontrollore e moduli per il comparto audio</a>	13
b. <a href="#">Conduzione ossea dell'audio</a>	14
i. <a href="#">Funzionamento</a>	14
ii. <a href="#">Pro e contro</a>	14
c. <a href="#">Modellazione e stampa 3D</a>	15
i. <a href="#">Sviluppo del modello 3D</a>	15
ii. <a href="#">Scelta del materiale di stampa</a>	15
d. <a href="#">Assemblaggio</a>	16
<b>Cap. 5 - <a href="#">Progettazione Software</a></b>	17
a. <a href="#">Calcolo della frequenza cardiaca e della temperatura corporea</a>	17
i. <a href="#">L'algoritmo</a>	17
ii. <a href="#">Codice</a>	19
b. <a href="#">Rilevamento delle cadute</a>	21
i. <a href="#">L'algoritmo</a>	21
ii. <a href="#">Codice</a>	23
c. <a href="#">Comunicazione con Arduino IoT Cloud</a>	25
i. <a href="#">Features</a>	25
ii. <a href="#">Connessione con gli occhiali</a>	25
<b>Cap. 6 - <a href="#">Raccolta ed analisi dei dati</a></b>	26
a. <a href="#">Metodologia</a>	26
b. <a href="#">Risultati e confronti con prodotti simili</a>	27
i. <a href="#">Rilevamento della frequenza cardiaca</a>	27
ii. <a href="#">Rilevamento delle cadute</a>	31
c. <a href="#">Conclusioni e prospettive di miglioramento</a>	37
<b>Cap. 7 - <a href="#">Conclusioni</a></b>	38
a. <a href="#">Dal prototipo al prodotto finito</a>	38
b. <a href="#">Possibili applicazioni future</a>	38
<b><a href="#">Ringraziamenti</a></b>	39
<b><a href="#">Bibliografia</a></b>	40



## Capitolo 1.

# Introduzione

Con l'avanzare tecnologico in epoca moderna, il campo dell'*IoT* (*Internet of Things*) sta crescendo a dismisura. Ad oggi siamo circondati da oggetti intelligenti e connessi in grado di raccogliere, elaborare ed inviare dati in maniera continua: uno studio condotto da *Statista* [1] e riportato nell'articolo di *FindStack* [2], alla fine del 2020 stimava 35.82 miliardi di dispositivi *IoT* connessi nel mondo. Nella maggior parte dei casi si tratta di piccoli microcontrollori dotati di sensori e modulo Wi-Fi, in altri casi si tratta di dispositivi con hardware molto più complessi.

L'applicazione dell'internet delle cose è estremamente variegata (domotica, dispositivi dotati di assistenti vocali, robotica, avionica<sup>1</sup>, telemetria, meteorologia, ecc.), ma in particolare nel campo biomedico ci si rivolge all'*IoT* in maniera sempre più assidua, grazie alla semplificazione del rapporto tra paziente e medico derivante dal suo utilizzo: prima dell'*IoT*, qualsiasi interazione medico-paziente doveva necessariamente avvenire tramite un classico incontro o una telefonata, non permettendo quindi al medico un monitoraggio costante e concreto del paziente anche a distanza.

L'avvento dei dispositivi *IoT*, quali smartwatch, fitness band ed altri in grado di effettuare molteplici misurazioni (tra cui ECG<sup>2</sup>, pressione arteriosa, glucosio nel sangue, ossigeno nel sangue ed altro) ha portato una maggiore consapevolezza dello stato di salute del proprio corpo ed un importante aiuto ai medici. I "Sensor-Equipped Eyeglasses" presentati nella tesi seguente si inseriscono nella branca biomedicale dei dispositivi *IoT*, integrando anche assistenza vocale e domotica.

Nel [Capitolo 2](#) vengono analizzati degli studi inerenti alle più comuni problematiche di salute dell'essere umano, con particolare attenzione alle fasce di età più avanzate. Inoltre, viene mostrato come degli occhiali intelligenti possano essere un ottimo strumento di prevenzione e ausilio alla risoluzione di molte di queste problematiche.

Nel [Capitolo 3](#) vengono descritte tutte le funzionalità attese e quelle che si intende implementare nel primo prototipo degli occhiali.

Nel [Capitolo 4](#) e nel [Capitolo 5](#) vengono esaminate tutte le scelte di progettazione hardware e software degli occhiali.

Nel [Capitolo 6](#) viene eseguita un'analisi dei dati raccolti durante le fasi di test del prototipo.

Nel [Capitolo 7](#) vengono infine affrontate le conclusioni dell'elaborato, con possibili applicazioni future del prodotto definitivo.

---

<sup>1</sup> Tecnologia applicata agli aeromobili ed al pilotaggio (autopilot, communication between aircraft ...)

<sup>2</sup> Elettrocardiogramma: riproduzione grafica dell'attività elettrica del cuore registrata a livello della superficie del corpo

## Capitolo 2.

# Studi, ricerche e definizione dell'approccio

### 2.a - Internet of Medical Things (IoMT).

L'importanza dell'*IoT* nel campo sanitario è tangibile al punto che ne è stata definita una categoria a parte: "Internet of Medical Things" abbreviato in "*IoMT*" oppure "*MIoT*".

Possiamo distinguere due principali categorie di dispositivi *IoT* nel campo biomedicale:

- dispositivi ad uso personale: in questa categoria ricadono tutti quei dispositivi facilmente reperibili dalle masse che consentono l'acquisizione di dati relativi alla salute fisica sfruttando sensori che, sebbene non siano pensati per un uso professionale, negli ultimi anni si sono rivelati in grado di produrre dati sempre più affidabili, anche grazie all'implementazione di algoritmi via via più sofisticati ed efficienti. I dispositivi ad uso personale (smartwatch, smart band, fasce cardio, etc.) permettono la produzione di dati inerenti alla frequenza cardiaca, ECG, ossigenazione del sangue, temperatura corporea, livello di glucosio nel sangue, pressione arteriosa, BIA<sup>3</sup>, rilevamento cadute e sono attualmente molto utilizzati per il monitoraggio durante lo svolgimento di attività fisica da parte del soggetto che se ne serve. Strumenti simili non si limitano alla raccolta passiva dei dati, essendo anche in grado di inviare avvisi di emergenza qualora dovessero rilevare condizioni critiche per la salute del soggetto: a tal proposito, ad oggi sono sempre maggiori i casi documentati di vite salvate proprio grazie all'utilizzo di questi dispositivi, che segnalano in tempo reale l'insorgere di patologie anche gravi, quali aritmie cardiache o apnee notturne, o permettono interventi tempestivi in caso di cadute critiche, allertando le competenti autorità sanitarie;
- dispositivi ad uso professionale, tra i quali rientrano i dispositivi professionali che permettono al medico il monitoraggio puntuale dei valori vitali del paziente; un tempo accessibili esclusivamente presso strutture sanitarie, questi strumenti consentono ora un monitoraggio anche da remoto, avvalendosi delle comunicazioni Internet e/o bluetooth. Appartengono a questa categoria, senza esaurirne la vasta gamma, i sensori ingeribili o impiantabili, che al servizio di raccolta dati possono accompagnare quello di somministrazione dei farmaci, gli activity trackers impiegati per effettuare un monitoraggio costante del soggetto durante il trattamento del cancro, o gli inalatori per asma connessi.

I "Sensor-Equipped Eyeglasses" appartengono alla categoria dei dispositivi ad uso personale degli IoMT. Il loro obiettivo è quello di offrire un tempestivo aiuto in caso si presentassero una o più condizioni critiche per la salute del soggetto che li utilizza. Il progetto è pensato primariamente per la fascia di età più avanzata della popolazione e per tutte le categorie fragili che possono beneficiare di un costante monitoraggio delle funzioni vitali.

---

<sup>3</sup> [Bioelectrical Impedance Analysis](#), secondo Wikipedia "è un metodo per stimare la composizione corporea, in particolare il grasso corporeo e la massa muscolare". Nel 2021 con l'uscita dello smartwatch Samsung® Galaxy Watch4, la casa coreana ha introdotto la BIA nei dispositivi indossabili.

## 2.b - Frequenza cardiaca ed ECG

La frequenza cardiaca è il numero di battiti del cuore al minuto (bpm) e rientra nelle funzioni vitali insieme alla temperatura corporea, alla pressione sanguigna e al ritmo respiratorio. Vedremo nei prossimi capitoli i metodi di calcolo più utilizzati.

Secondo uno studio di *Eurostat* [3] nel 2016 circa l'82,9% dei decessi verificatisi nell'Unione Europea sono attribuibili a soggetti di età pari o superiore a 65 anni. Il 40,3% dei decessi avvenuti tra la popolazione anziana sono in più dovuti a malattie circolatorie, vale a dire cardiopatie ischemiche, tra le quali rientrano gli infarti, e le malattie cerebro vascolari come gli ictus (figura 2.1).

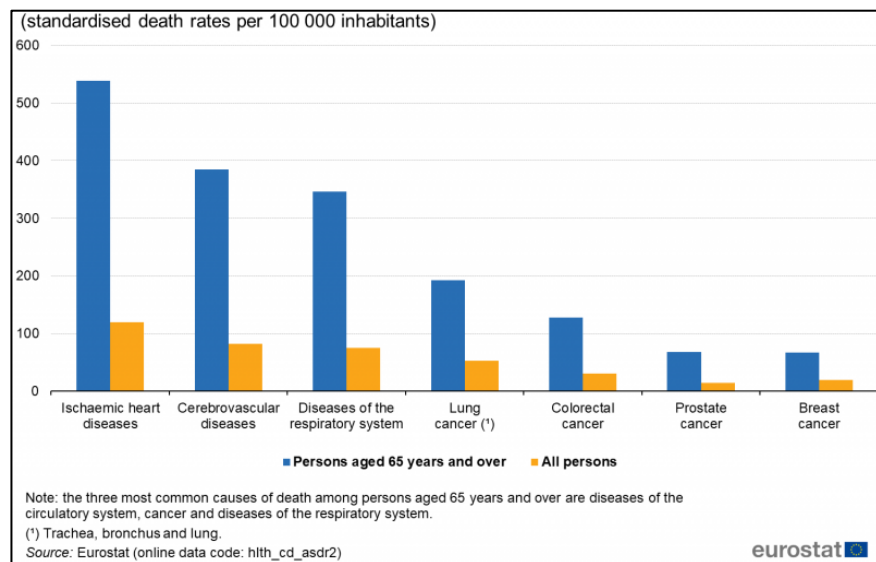


Figura 2.1 - Major causes of death, EU, 2016

Per la prevenzione di patologie di questo tipo, l'ECG [4] ed il monitoraggio della frequenza cardiaca risultano essere strumenti essenziali.

Il funzionamento è molto semplice: i dispositivi *wearable*<sup>4</sup> in grado di eseguire un ECG sono dotati di due elettrodi che misurano la differenza di potenziale elettrico prodotto dalle singole cellule muscolari del nostro cuore. Gli impulsi elettrici generati da ogni battito si trasmettono attraverso il corpo grazie alla conducibilità del liquido interstiziale<sup>5</sup> fino a raggiungere gli elettrodi del sensore che permettono il tracciamento dell'elettrocardiogramma attraverso lo studio delle onde rilevate e degli intervalli di tempo tra esse intercorrenti (figura 2.2).

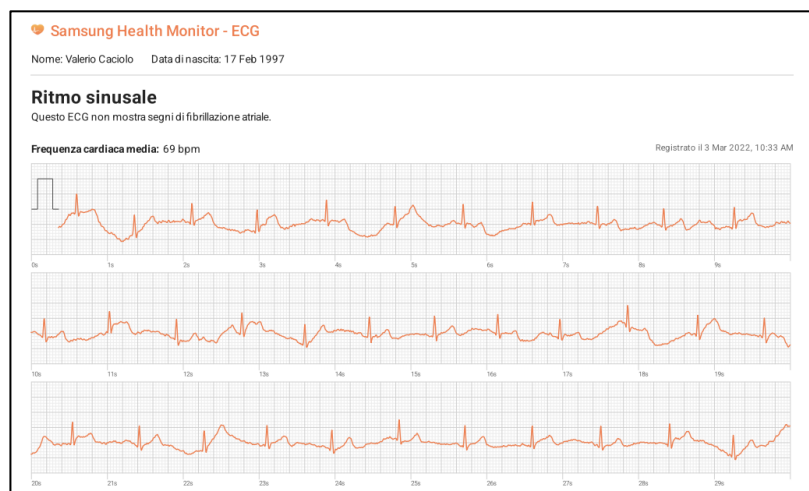


Figura 2.2 - Tracciato di un ECG (eseguito con uno smartwatch Samsung Galaxy Watch 4)

<sup>4</sup> Dispositivi indossabili, come smartwatch, smartband, etc.

<sup>5</sup> Secondo Wikipedia "In istologia, per liquido interstiziale si intende la soluzione acquosa presente fra le cellule di un tessuto"

## 2.c – Rischio da caduta

Un importante studio condotto dalla *World Health Organization* [5] afferma che il 28-35% delle persone nella fascia di età superiore ai 65 anni cade in media 2-4 volte l'anno, ma la percentuale aumenta al 32-42% per gli anziani over 70, che cadono in media 5-7 volte l'anno. La frequenza delle cadute, così come l'incidenza di cadute fatali (figura 2.3), aumenta in base all'età ed al livello di fragilità dell'individuo.

I fattori di rischio per una persona anziana che possono portare al verificarsi di una caduta accidentale sono molteplici (figura 2.4) e, considerando l'alta probabilità di una caduta grave, la tempestività dei soccorsi diviene di vitale importanza: infatti, anche non considerando i casi in cui la caduta causi la perdita dei sensi, recenti studi [6] dimostrano come il 47% delle persone anziane non sia in grado di rialzarsi dopo una caduta accidentale e non riesca pertanto a chiedere aiuto. Non sono rari i casi documentati di anziani rinvenuti senza vita a causa dell'aggravarsi dei danni comportati da una caduta accidentale a seguito della quale i soggetti, impossibilitati a muoversi, non sono riusciti ad allertare i soccorsi. Morti simili sarebbero potenzialmente evitabili se i singoli fossero in grado di ottenere aiuti tempestivi.

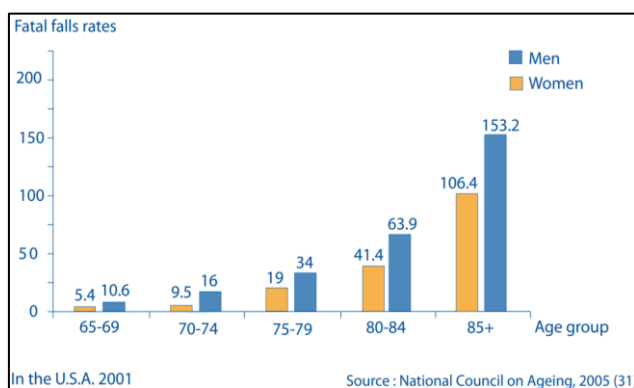


Figura 2.3 – Fatal falls rate by age and sex group

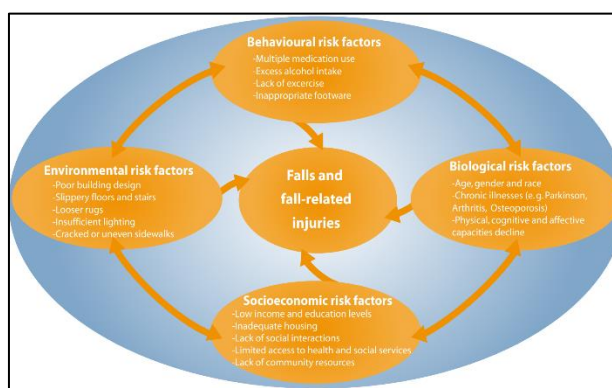


Figura 2.4 – Risk factor model for falls in older age

## 2.d – Importanza del pronto intervento

Nei due paragrafi precedenti sono state evidenziate due tra le più comuni situazioni di pericolo per la salute di una persona anziana. Comun denominatore delle situazioni illustrate è l'importanza di un soccorso immediato. Se è vero che la rilevazione tempestiva di eventuali cardiopatie tramite l'elettrocardiogramma e il monitoraggio della frequenza cardiaca possono essere decisivi nel prevenire l'insorgere di malattie collegate all'apparato cardiocircolatorio, è altrettanto vero che allertare in tempo i soccorsi a seguito di una caduta grave può fare la differenza tra la vita e la morte.

## 2.e – Come poter monitorare i dati

Sul mercato esistono attualmente molte tipologie di dispositivi wearable: orologi, bracciali, anelli, polsini, cappelli, etc. Il problema principale connesso all'utilizzo di questi strumenti da parte di un anziano è la non essenzialità dell'oggetto: un orologio o un anello non sono elementi ricorrenti nella quotidianità di una persona anziana e questo comporta spesso una certa noncuranza da parte del soggetto, che può quindi tendere a dimenticare il dispositivo invece di indossarlo.

Secondo uno studio di Statista [7] eseguito nel 2016 in America il 77,5% degli over 55 utilizza un paio di occhiali. È facile supporre che questa percentuale aumenti proporzionalmente con l'avanzare dell'età. Si può giungere quindi alla conclusione che un paio di occhiali possa essere un'ottima base per l'implementazione di un sistema di monitoraggio dedicato ad un anziano: non solo sono essenziali per molti dei soggetti interessati, ma possono essere comunque utilizzati anche da chi non ha problemi di vista applicando lenti neutre alla montatura. Inoltre, permetterebbero di unire diverse features come funzioni di riproduzione audio (grazie alla loro vicinanza con l'apparato uditivo) e di conseguenza la possibilità di implementare un assistente vocale.



## Capitolo 3.

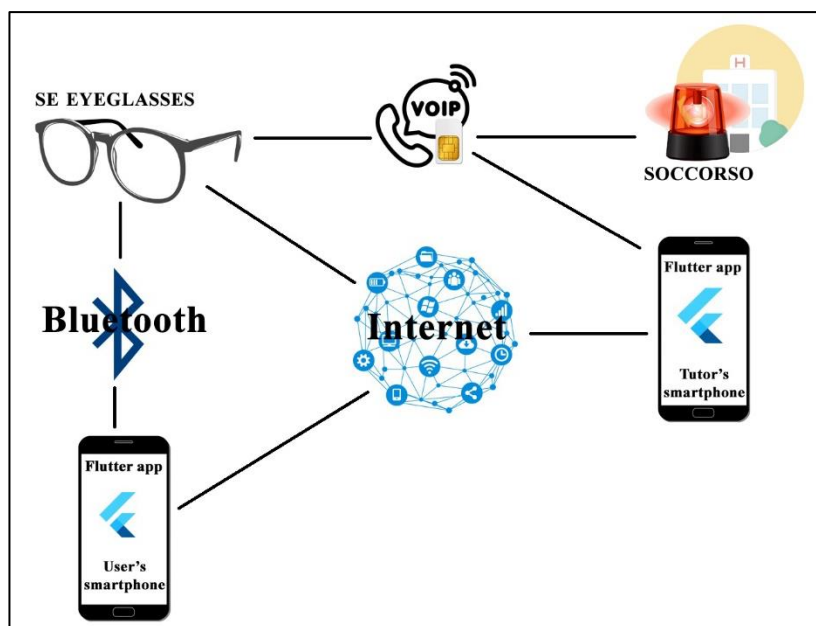
# Funzionalità

In questo capitolo verrà esposta un'analisi del contesto d'uso degli "SE Eyeglasses", per poi presentare i vari requisiti, funzionali e non, del progetto e, in ultimo, le funzionalità scelte per lo sviluppo di un prototipo.

### 3.a – Analisi del contesto

Gli "SE Eyeglasses" comunicano via Internet e bluetooth con lo smartphone dell'anziano e/o di un tutore e sono in grado di effettuare chiamate di emergenza (figura 3.1). L'utilizzatore degli occhiali e/o un suo tutore ha/hanno quindi la possibilità di accedere a tutti i parametri vitali misurati e alle altre informazioni relative al dispositivo (come lo stato di carica) tramite l'utilizzo di un'applicazione mobile.

Gli occhiali sono utilizzabili anche come dispositivo stand-alone e non necessiteranno dunque obbligatoriamente di connettersi ad un eventuale smartphone dell'anziano<sup>6</sup>.



**Figura 3.1** – Analisi di contesto SE Eyeglasses [\[9\]](#)

L'utilizzo stand-alone degli occhiali prevede però comunque una connessione ad Internet affinché sia possibile implementare le funzionalità presentate nel prossimo paragrafo.

Gli occhiali possono collegarsi in diversi modi alla rete:

- utilizzando una SIM (grazie allo slot dedicato) o una e-SIM di un operatore telefonico, consentendo attraverso la rete dati 4G l'inoltro di notifiche di emergenza ai contatti scelti o semplicemente l'effettuazione di telefonate o l'impiego dell'assistente vocale;
- utilizzando una connessione bluetooth con uno smartphone, consentendo grazie a quest'ultimo l'inoltro di notifiche di emergenza ai contatti scelti o semplicemente l'effettuazione di telefonate o l'impiego dell'assistente vocale;
- utilizzando una connessione WiFi, che a differenza della connessione via SIM e via smartphone tramite bluetooth, non consente di effettuare chiamate tradizionali.

<sup>6</sup> L'80% degli over 65 possiede un telefono. Solo il 42% di questi possiede uno smartphone [\[8\]](#)

### 3.b – Requisiti

Il capitolo seguente illustra tutti i requisiti attesi del progetto, funzionali<sup>7</sup> e non<sup>8</sup>.

Tra i requisiti funzionali sono stati evidenziati in **grassetto** quelli effettivamente implementati nel prototipo presentato nei paragrafi seguenti, ed evidenziati in **celeste** quelli implementabili in un'eventuale versione futura.

#### 3.b.i – Funzionali

##### Ricaricare il dispositivo:

*Il dispositivo è in grado di essere ricaricato tramite l'uso di un cavo o una base magnetica ad induzione.*

- **Eseguire una ricarica con cavo;**
- **Eseguire una ricarica ad induzione wireless.**

##### Misurare i parametri vitali e non:

*Il dispositivo permette di ottenere i valori numerici di diversi parametri vitali oltre alla possibilità di esecuzione di un elettrocardiogramma.*

- **Misurare la frequenza cardiaca;**
- **Misurare la temperatura corporea;**
- **Misurare ossigeno nel sangue;**
- **Misurare temperatura ambiente;**
- **Eseguire un ECG.**

##### Monitoraggio passivo degli occhiali:

*Il dispositivo esegue in background un continuo monitoraggio dei parametri vitali e dei valori dell'accelerometro e giroscopio integrati ai fini di rilevare situazioni anomale e/o pericolose per la salute dell'utilizzatore.*

- **Monitoraggio di cadute;**
- **Monitoraggio battito cardiaco;**
- **Monitoraggio della temperatura corporea;**
- **Monitoraggio del livello di ossigeno nel sangue;**
- **Monitoraggio presenza di fumo o gas nell'aria.**

##### Invio di avvisi di emergenza:

*Il dispositivo è in grado di inviare in modo autonomo degli avvisi di emergenza (ai soccorsi o ai contatti scelti) generati a seguito del verificarsi di condizioni pericolose per la salute dell'utilizzatore.*

- **Invio avviso di avvenuta caduta passivo;**
- **Invio avviso di criticità cardiache passivo;**
- **Invio avviso di criticità ambientali;**
- **Invio avviso di emergenza manuale.**

##### Visualizzare i valori dei parametri vitali e degli avvisi tramite app mobile;

*L'utilizzatore del dispositivo e/o un suo tutore è/sono in grado di accedere alle varie informazioni (parametri vitali misurati, impostazioni, informazioni sullo stato degli occhiali ecc.) attraverso l'impiego di un'applicazione mobile per Android, iOS e HarmonyOs.*

##### Gestire la multimedialità:

*L'utilizzatore del dispositivo è in grado di accedere ad una vasta gamma di funzionalità multimediali attraverso l'interazione con gli occhiali e grazie all'hardware degli stessi.*

- **Interagire con l'assistente vocale;**
- **Ascoltare la musica;**
- **Effettuare e ricevere chiamate;**
- **Controllare dispositivi domotici;**
- **Sentire delle istruzioni vocali per la configurazione iniziale.**

#### 3.b.ii – Non funzionali

##### Prestazioni ed efficienza:

- velocità ed affidabilità di inoltro di un avviso di emergenza: vista l'importanza degli avvisi di emergenza, il sistema deve garantire un rapido inoltro in tutta sicurezza dell'avviso;
- il sistema deve garantire una autosufficienza energetica che assicuri almeno una giornata intera di utilizzo;
- precisione ed affidabilità delle stime numeriche dei parametri vitali con minimo margine di errore.

<sup>7</sup> Sono tutte quelle funzionalità o servizi che gli occhiali dovranno fornire

<sup>8</sup> Sono tutti quei vincoli e proprietà che gli occhiali dovranno soddisfare in ambito di sicurezza, affidabilità...

**Usabilità:**

- visto il target del prodotto, il sistema deve essere estremamente user-friendly in tutte le sue componenti.

**Affidabilità:**

- il prodotto deve garantire la capacità di rispettare tutti i requisiti funzionali. Le operazioni legate ad un requisito non devono influenzare quelle legate agli altri requisiti;
- il prodotto deve essere resistente agli urti ed alle sollecitazioni;
- il prodotto deve garantire una buona copertura della rete mobile tramite Sim o e-Sim.

**Portabilità:**

- il progetto deve assicurare totale compatibilità lato software con tutti i sistemi operativi più famosi e diffusi così da garantire il funzionamento del prodotto in ogni condizione.

**Supportabilità:**

- il prodotto deve essere aggiornato regolarmente nel tempo per rispettare degli standard di sicurezza e prestazioni.

**Adattabilità:**

- il prodotto deve resistere alle diverse condizioni atmosferiche (invernali ed estive);
- il prodotto deve avere una forma adattabile alla maggior parte dei volti delle persone.

**3.c – Il prototipo**

Per dimostrare le potenzialità del progetto è stato creato un prototipo degli occhiali che implementa i requisiti funzionali evidenziati in **grassetto** nei paragrafi precedenti e prende come riferimento i requisiti non funzionali per cercare di avvicinare il prototipo al prodotto finito.

Le funzionalità scelte per il prototipo sono state il rilevamento della frequenza cardiaca e della temperatura corporea, il rilevamento delle cadute, gli avvisi di emergenza e la parte multimediale.

Nei prossimi capitoli viene affrontato lo sviluppo del solo prototipo.

## Capitolo 4.

# Progettazione hardware

### 4.a – Scelta dell'hardware

Al fine di implementare i requisiti funzionali e non funzionali descritti nel Capitolo 3 nel prototipo degli occhiali sono state utilizzate delle componenti facilmente reperibili e a basso costo. Lo schema in figura 4.1 mostra l'architettura hardware degli occhiali. Sono stati impiegati due microcontrollori<sup>9</sup> (uno per asta): uno dedicato al monitoraggio ed alle comunicazioni via Internet, l'altro riservato alla componente audio e di assistenza.

La necessità di utilizzo di due diversi microcontrollori per il prototipo è dovuta alla difficile reperibilità di PCB<sup>10</sup> complete e ad alte prestazioni nel mercato. Le case produttrici di wearable disegnano e producono internamente queste schede che non sono disponibili all'acquisto da parte di utenti privati. Il tutto è stato alimentato da due batterie Lipo da 3.7v di capienza 300mAh con possibilità di essere ricaricate tramite cavo. In un'ottica futura gli "SE Eyeglasses" disporranno di un singolo microcontrollore, prodotto su misura, che si occuperà di tutte le funzionalità

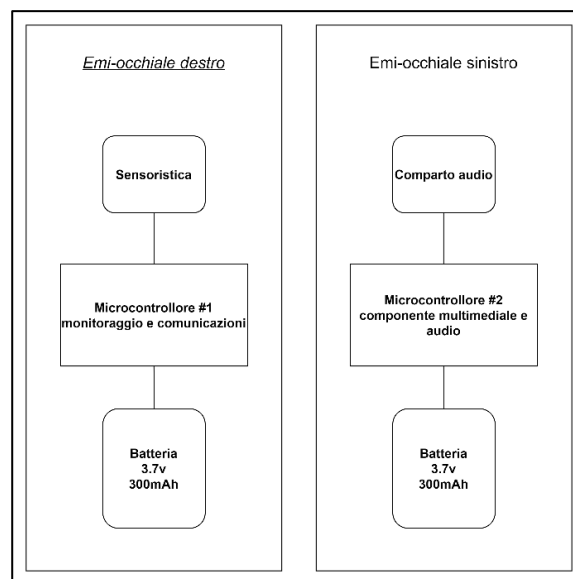


Figura 4.1 – Architettura hardware (created with [draw.io](https://draw.io))

#### 4.a.i – Emi-occhiale destro: Arduino e sensori

Arduino è un noto produttore di software e piattaforme hardware dotate di microcontrollore che consentono di sviluppare e dare forma ai più disparati progetti. Le loro schede elettroniche si rivelano estremamente versatili grazie alla loro architettura hardware che mette a disposizione del progettista diversi ingressi analogici e digitali completamente programmabili tramite la scrittura di codice in linguaggio C++. Le schede sono inoltre dotate di pin di ingresso e uscita di corrente a 3.3v e 5v per alimentare la totalità dei sensori. La piattaforma hardware scelta per gli "SE Eyeglasses" è stata la scheda **Arduino® nano 33 IoT** [10] (figure 4.2, 4.3a, 4.3b). Essa è dotata di:

- un microcontrollore SAMD21 Cortex®-M0++ 32 bit low power ARM MCU che gira fino a 48MHz;
- memoria CPU flash da 256KB;
- memoria RAM da 32KB;
- un modulo NINA-W10 con WiFi low energy 2.4GHz e Bluetooth 4.1;
- un modulo LSM6DSOXTR IMU<sup>11</sup> a 6 assi.

La scheda presenta 9 pin analogici e 14 digitali. Supporta il sistema di comunicazione seriale I<sup>2</sup>C ed opera ad un voltaggio di 3.3V.

<sup>9</sup> Secondo Wikipedia "Un [microcontrollore](https://en.wikipedia.org/wiki/Microcontroller) (in acronimo MCU ovvero MicroController Unit), in elettronica digitale, è un dispositivo elettronico integrato su singolo circuito elettronico, utilizzato generalmente in sistemi embedded".

<sup>10</sup> Printed circuit board, componente utilizzato per interconnettere i vari componenti elettronici di un circuito.

<sup>11</sup> Inertial measurement unit, un sistema avionico basato su sensori inerziali, come accelerometri e giroscopi.

La scheda Arduino si occupa della parte “health” degli occhiali. Il modulo LSM6DSOXTR permette di raccogliere dati inerenti il giroscopio e l’accelerometro, che tramite un apposito algoritmo, permettono di rilevare il verificarsi di un’eventuale caduta.

Il modulo WiFi/BLE NINA-W10 consente invece la comunicazione dei dati ad un cloud dedicato cui, in questo primo prototipo, è assegnato anche il compito di inoltrare segnalazioni di emergenza ai contatti scelti a seconda delle preferenze dell’utente.

Il rilevamento della frequenza cardiaca, dell’ossigenazione del sangue e della temperatura sono demandati al sensore **MAX30102** (figura 4.4). Il Maxim Integrated Max30102 è un pulsossimetro e sensore di frequenza cardiaca che include due diodi ad emissione luminosa (uno a luce rossa o verde ed uno a luce infrarossa), un fotorelevatore e un termometro, il tutto implementato con dell’elettronica a basso rumore che sfrutta la comunicazione seriale  $I^2C$ .

Per effettuare le misurazioni, il MAX30102 usa un metodo chiamato Photoplethysmography<sup>12</sup>[11]: attraverso il led, il sensore diffonde la luce sulla pelle e riesce a misurare la perfusione<sup>13</sup> nel sangue, distinguendola da quella tissutale ed ossea.

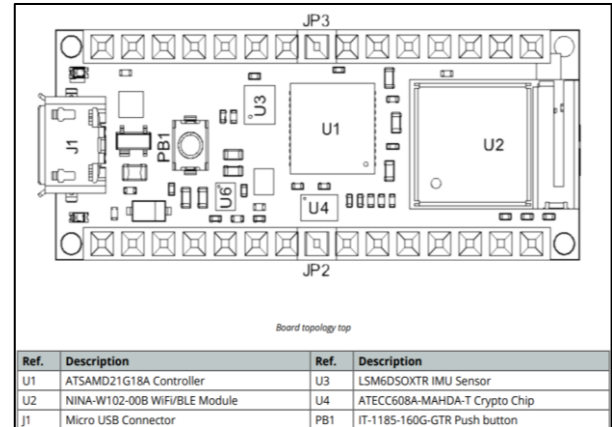


Figura 4.2 – Board Topology [10]

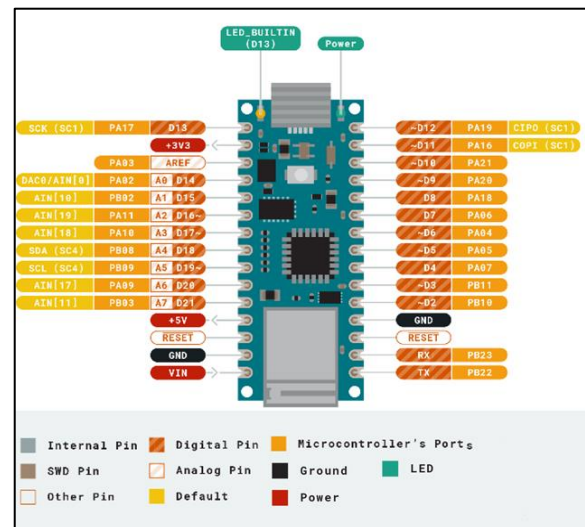


Figura 4.3a – Board Pinouts [10]

La luce riflessa dal sangue (ciascuna lunghezza d’onda) viene catturata dal fotodiodo e convertita in corrente, la quale genera quindi un dato interpretabile per la misurazione della variazione della circolazione sanguigna.

Il dato grezzo permette di misurare la frequenza cardiaca e l’ossigenazione del sangue del soggetto con l’uso di algoritmi in grado di calcolarne i valori.

La tecnica della fotoplethysmografia è utilizzata dalla maggior parte dei dispositivi wearable disponibili in commercio.

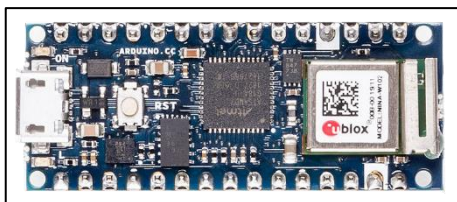


Figura 4.3b – Arduino Nano 33 IoT Board

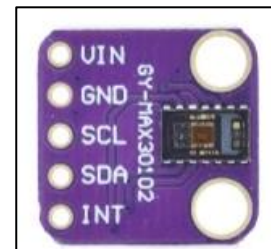


Figura 4.4 – MAX30102 (Da [Joom](#))

<sup>12</sup> La [photoplethysmography](#) è lo studio della diffusione dei raggi infrarossi nei tessuti, per studiarne l'irrorazione (Da Treccani)

<sup>13</sup> In fisiologia, la [perfusione](#) è il processo in cui un corpo fornisce il sangue per il letto capillare sin nel tessuto. (Da Wikipedia)

Come già specificato, il collegamento tra il microcontrollore ed il sensore avviene tramite comunicazione seriale  $I^2C$  come mostrato nel seguente diagramma (figura 4.5):

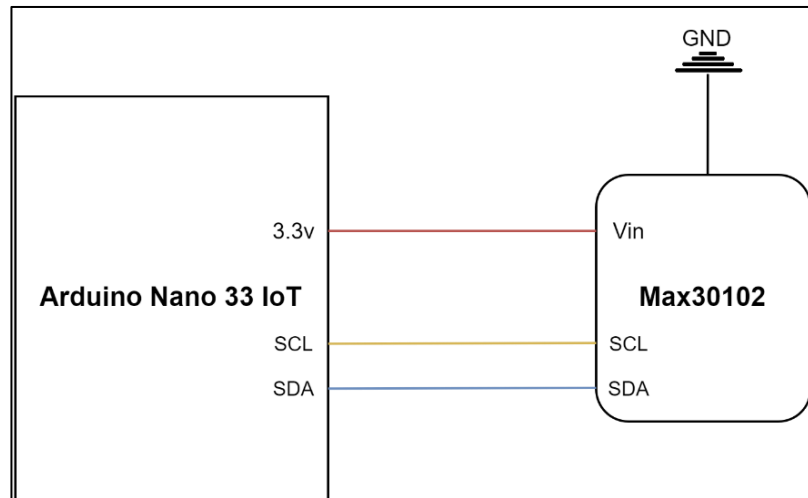


Figura 4.5 – Circuit Design DX (created with [draw.io](https://draw.io))

#### 4.a.ii – Emi-occhiale sinistro: microcontrollore e moduli per il comparto audio

Per la componente audio si è scelto di utilizzare come microcontrollore una scheda Mencom GT 801, una comune scheda audio dotata di comunicazione Bluetooth con microfono. Sono stati poi utilizzati un modulo amplificatore PAM8403 a 3Ω e due trasduttori audio, il GD12 e il GD01 (figure 4.6, 4.7, 4.8, 4.9).



Figura 4.6 – Mencom GT 801

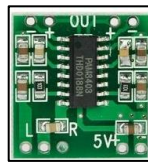


Figura 4.7 – PAM8403

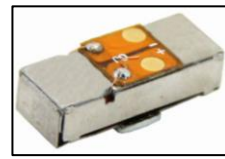


Figura 4.8 – GD12



Figura 4.9 – GD01

Grazie a queste componenti gli occhiali sono in grado di collegarsi tramite Bluetooth ad uno smartphone che funge da sorgente audio. L'audio viene poi riprodotto tramite conduzione ossea attraverso i due trasduttori, di cui viene affrontato il funzionamento nei seguenti paragrafi.

La progettazione del circuito è mostrata dal seguente diagramma (figura 4.10):

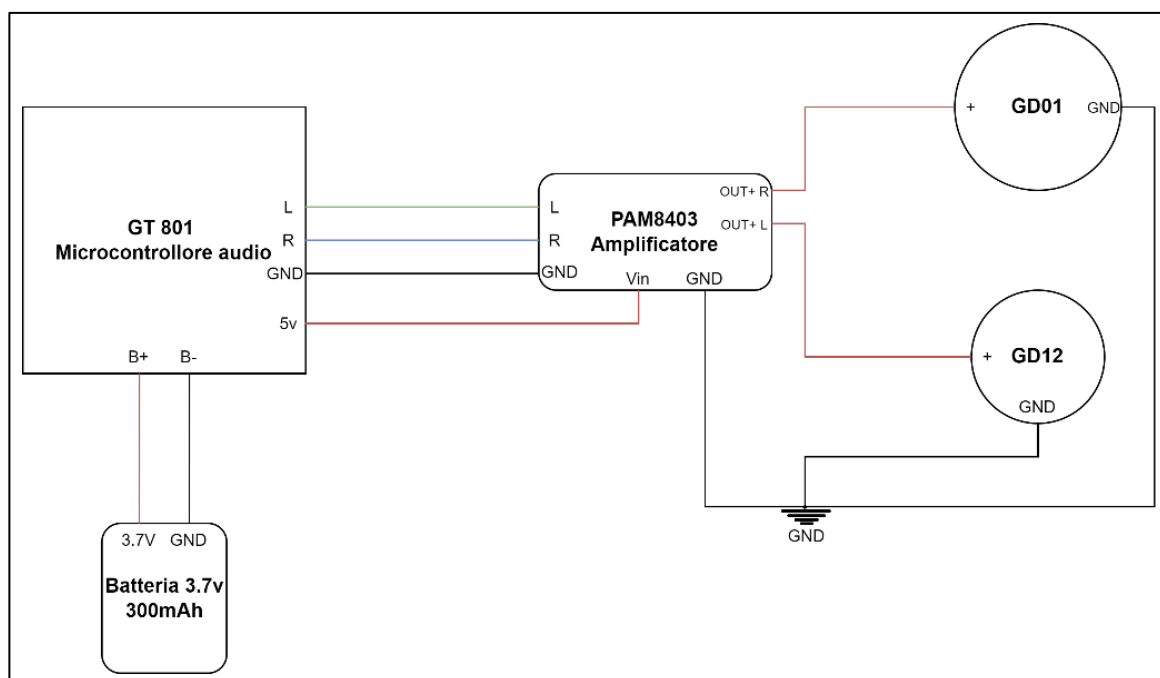


Figura 4.10 – Circuit Design SX (created with [draw.io](https://draw.io))



## 4.b – Conduzione ossea dell'audio

Gli "SE Eyeglasses" utilizzano la tecnologia di conduzione ossea per veicolare l'audio dell'assistente vocale, delle chiamate e in generale di tutte quelle funzionalità che prevedono la riproduzione di contenuti audio. In questo breve paragrafo viene illustrato il funzionamento ed i vantaggi/svantaggi di questo genere di tecnologia.

### 4.b.i – Funzionamento

Come riportato sul sito di *Goldendance* [12], nella maggior parte dei casi l'essere umano è in grado di percepire i suoni trasmessi nell'aria attraverso i timpani (figura 4.11). Il timpano converte le onde sonore in vibrazioni e le trasmette alla coclea<sup>14</sup>. Tuttavia, alcuni suoni, di cui la voce umana ne è un esempio, vengono percepiti senza passare per il timpano: ciò accade poiché la nostra voce raggiunge la coclea grazie alla trasmissione da parte di ossa e tessuti.

La veicolazione audio tramite conduzione ossea sfrutta i trasduttori, dispositivi in grado di svolgere la stessa funzione del timpano convertendo i suoni in vibrazioni interpretabili dalla coclea (figura 4.12).

### 4.b.ii – Pro e contro

Analizziamo in breve i vantaggi e gli svantaggi di questo approccio:

- i vantaggi sono molteplici soprattutto per la fascia di età a cui gli "SE Eyeglasses" sono indirizzati. Il timpano resta libero di ascoltare altri suoni provenienti dall'ambiente esterno, caratteristica fondamentale per un anziano affinché non abbia problemi di equilibrio (le comuni cuffie, inserendosi nell'orecchio, ostacolano il normale percepimento dei suoni e possono causare i problemi suddetti). Inoltre, come anticipato precedentemente, l'ascolto di suoni tramite conduzione ossea bypassa l'uso dei timpani: la "Conductive Hearing Loss", una condizione molto comune nella fascia avanzata di età, è associata ad uno scarso funzionamento del timpano che determina una parziale perdita della capacità uditiva. Grazie alla conduzione ossea anche un soggetto affetto da una condizione simile è in grado di sentire; non a caso i più famosi rimedi tecnologici alla perdita dell'udito consistono proprio nell'utilizzo della conduzione ossea per amplificare i suoni esterni catturati tramite un microfono ad alta sensibilità;
- il principale svantaggio della conduzione ossea è la qualità dell'audio, non paragonabile ad uno speaker tradizionale. Nel progetto viene utilizzato un amplificatore per cercare di colmare il divario, soprattutto relativamente al volume massimo raggiungibile dall'audio. Vista la natura del prodotto, non pensato per un ascolto multimediale ad alta fedeltà, questo difetto non rappresenta però un ostacolo vero e proprio all'implementazione di una simile tecnologia.

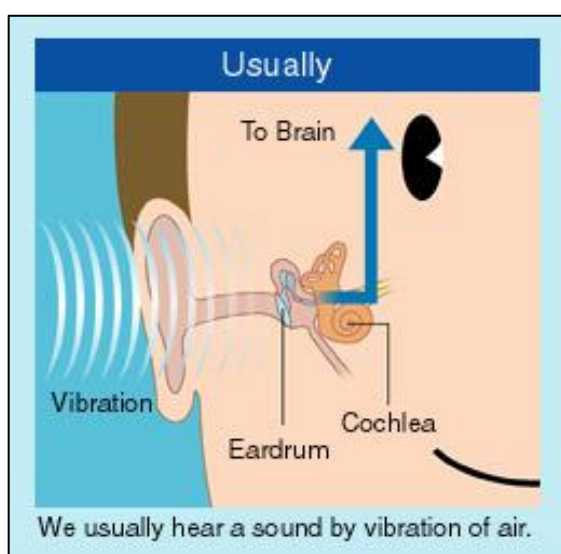


Figura 4.11 – Ascolto tramite timpano [12]

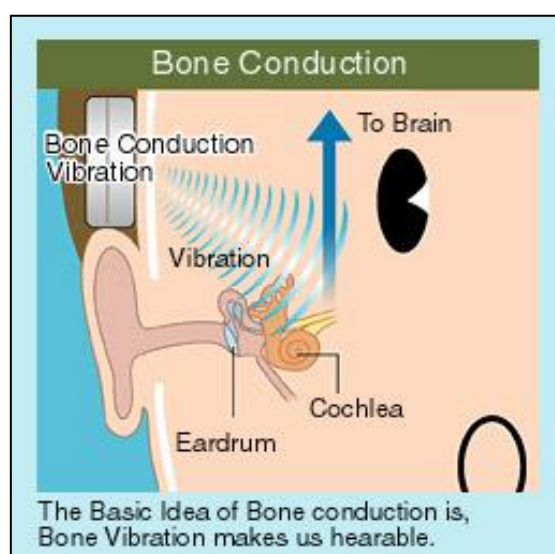


Figura 4.12 – Ascolto tramite conduzione [12]

<sup>14</sup> Coclea o orecchio interno: si occupa di tradurre l'informazione acustica appena ricevuta dal timpano in impulsi nervosi in modo da renderla comprensibile al nostro cervello

## 4.c – Modellazione e stampa 3D

Lo sviluppo e la produzione del prototipo sono avvenuti grazie all'uso della modellazione e della stampa 3D.

### 4.c.i – Sviluppo del modello 3D

Per lo sviluppo di un modello stampabile da una qualsiasi stampante 3D cartesiana è stato utilizzato il software "Autocad Fusion 360"<sup>15</sup>. Le figure 4.13, 4.14 e 4.15 mostrano le varie versioni del modello fino a quella definitiva di test per il prototipo, di cui è possibile visionare un timelapse della produzione e della stampa al seguente link su [GitHub](#). Studiato per essere completamente modulare in modo da facilitarne la modifica delle varie parti che lo compongono, il modello è pensato specificatamente per le componenti precedentemente descritte. Nel corpo centrale dell'emi-occhiale destro è stata alloggiata la scheda Arduino Nano 33 IoT (figura 4.3b), da cui partiranno 4 cavi fino al terminale<sup>16</sup> in cui è stato alloggiato il Max30102 (figura 4.4). Vista la struttura, i terminali dell'occhiale sono posizionati esattamente dietro il lobo dell'orecchio, rendendo possibile la misurazione del battito cardiaco da parte del Max30102. Nel corpo centrale dell'emi-occhiale sinistro sono stati invece alloggiati il GT 801 (figura 4.6) ed il PAM8403 (figura 4.7); da quest'ultimo partono due coppie di cavi collegate ai trasduttori GD01 e GD12 (figure 4.8, 4.9), il primo posizionato sul terminale dell'occhiale a diretto contatto con il processo mastoideo<sup>17</sup>, il secondo posizionato all'inizio dell'asta e quindi a contatto con la tempia.



Figura 4.13 – Model rendering SE Eyeglasses\_1.0v



Figura 4.14 – Model rendering SE Eyeglasses\_2.0v



Figura 4.15 – Model rendering SE Eyeglasses\_3.0v

### 4.c.ii – Scelta del materiale e stampa

Il materiale impiegato per la stampa del prototipo è stato il PLA (poly lactic acid), un polimero compostabile e biodegradabile prodotto dalla trasformazione degli zuccheri estratti da risorse rinnovabili come l'amido di mais, bagassa di canna da zucchero ed altri prodotti ad alto contenuto di amido.

Si è scelto di utilizzare il PLA per diversi motivi:

- il basso costo di produzione e di conseguenza di acquisto (1kg  $\cong$  20€);
- il fatto che durante il processo di stampa non vengono prodotti fumi tossici come avviene invece durante la stampa di materiali quali ABS<sup>18</sup> o resina;
- la bassa temperatura di fusione (180 °C);
- la buona resistenza ed elasticità.

Un difetto non indifferente di questo materiale bioplastico è però riscontrabile nella sua bassa resistenza al calore: esposto a temperature superiori ai 50 °C il PLA tende a deformarsi e a sfaldarsi. In un'ottica futura gli "SE Eyeglasses" utilizzeranno materiali più resistenti, come il prima citato ABS o l'alluminio.

<sup>15</sup> [Sito del produttore](#)

<sup>16</sup> Parte finale dell'asta di un occhiale che si va a posizionare dietro l'orecchio

<sup>17</sup> Componente del cranio di un essere umano ([immagine](#))

<sup>18</sup> Acrilnitrile butadiene stirene è un materiale termoplastico amorfo



#### 4.d – Assemblaggio

Dopo la rifinitura del modello 3D a seguito delle prime stampe e dopo aver tentato diverse configurazioni progettate per ottenere quella più congeniale allo scopo, si è giunti alla stampa 3D del modello definitivo per il prototipo. (figura 4.16).

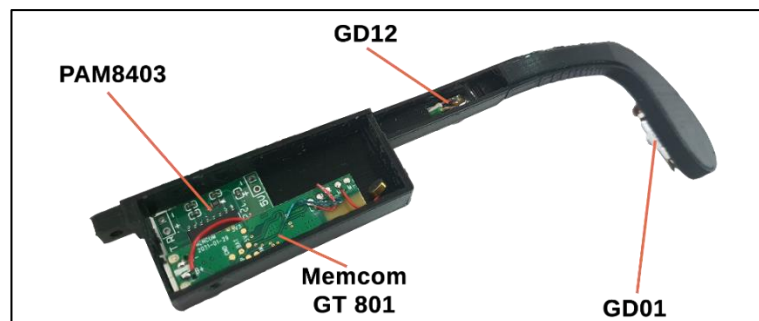


**Figura 4.16** – Model rendering della versione definitiva del prototipo

Le figure 4.17, 4.18 riportano il prototipo degli “SE Eyeglasses”, focalizzandosi in particolare sulle aste degli occhiali e sui loro componenti.



**Figura 4.17** – asta destra prototipo “SE Eyeglasses”



**Figura 4.18** – asta sinistra prototipo “SE Eyeglasses”

La figura 4.19 riporta infine il prototipo completo comprensivo di tutte le componenti hardware.



**Figura 4.19** – Prototipo degli “SE Eyeglasses”

I file “.stl” dei modelli sono disponibili al link [GitHub](#)

## Capitolo 5.

# Progettazione software

### 5.a – Calcolo della frequenza cardiaca e della temperatura corporea

In questo paragrafo viene presentato l'algoritmo responsabile del calcolo della frequenza cardiaca e della temperatura corporea utilizzando i dati raccolti del sensore MAX30102 precedentemente descritto.

#### 5.a.i – L'algoritmo

L'algoritmo utilizza la libreria "MAX30105.h" [\[13\]](#) per la lettura del dato grezzo prodotto dal sensore e della libreria "Wire.h"<sup>19</sup> per la comunicazione seriale I<sup>2</sup>C.

Il dato grezzo prodotto dal sensore, definito valore IR, viene interpretato dalla libreria MAX30105.h che, attraverso la funzione "bool checkForBeat(int32\_t sample)", segnala l'occorrenza di un battito del cuore.

Questa funzione è in grado di rilevare i picchi del valore IR (figura 5.1), ovvero i momenti in cui il dato subisce una rapida diminuzione del valore dopo una precedente crescita: ogni picco corrisponde ad un battito cardiaco. Rilevato ciascun battito, viene salvato il tempo in cui è avvenuto e quindi calcolato il tempo intercorso tra i battiti per stilare una prima stima della frequenza cardiaca. Le varie stime calcolate sono di volta in volta salvate in una pila di dieci elementi, la cui media aritmetica ci fornisce una valutazione accettabile della frequenza cardiaca in tempo reale.

La temperatura è letta tramite la funzione "float readTemperature()" che legge i dati grezzi del termometro integrato nel sensore e ne riporta il valore in gradi Celsius.

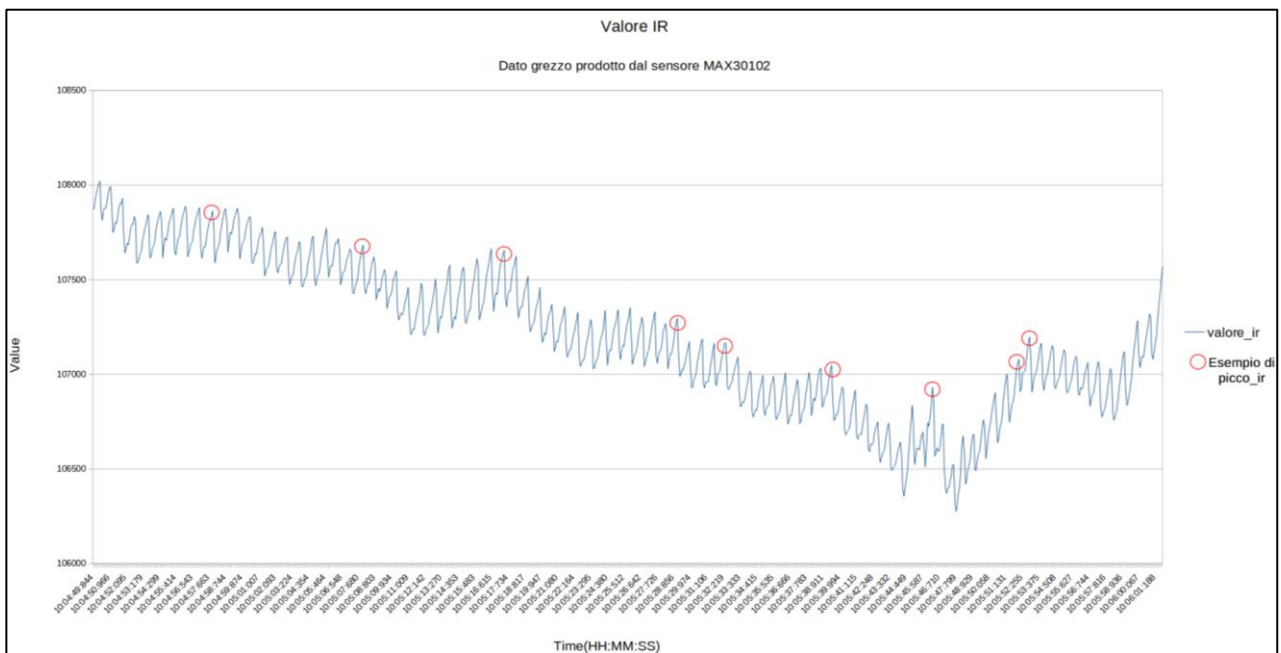


Figura 5.1 – grafico misurazione del valore IR grezzo dal MAX30102

<sup>19</sup> Libreria ufficiale di Arduino per l'utilizzo di moduli a comunicazione seriale. [Reference](#)

Come si può evincere dalla figura 5.2, gli occhiali sono finora in grado di fornire dati come la temperatura corporea (in gradi Celsius) e la frequenza cardiaca; nella versione finale del codice sarà l'analisi di dati simili a generare l'inoltro di notifiche di allerta in caso di rilevamento di valori anomali.

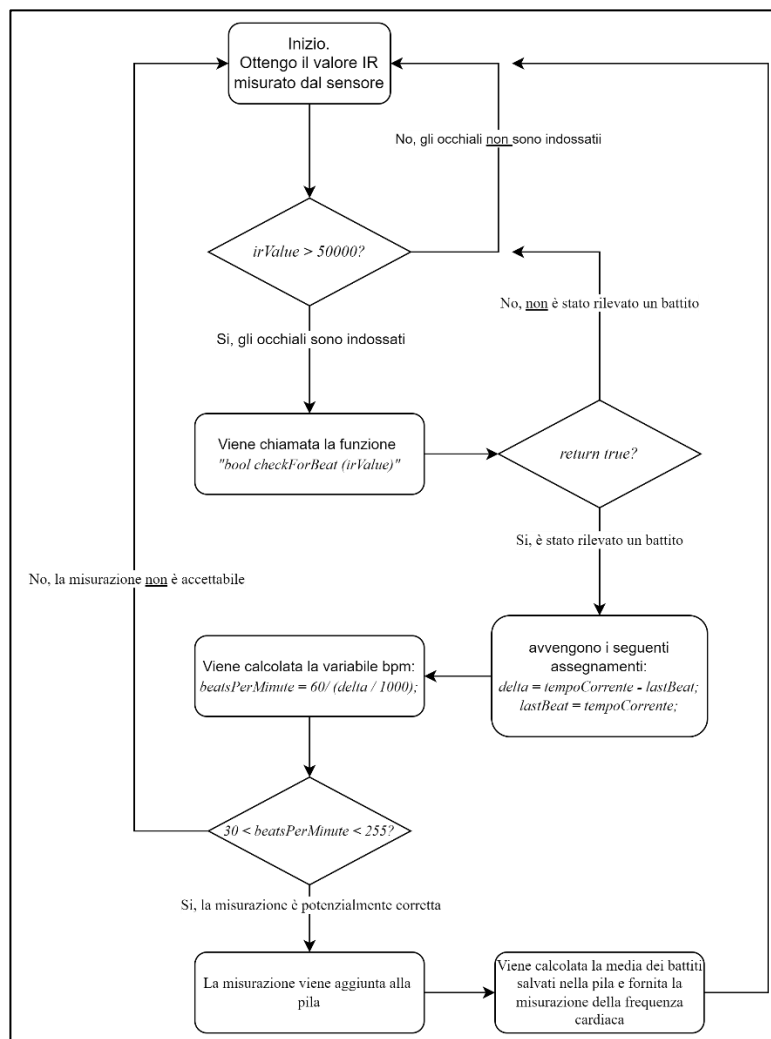
```

11:47:32.335 -> temperatureC=36.6250, IR=95088, INSTA_BPM=81.30, Avg BPM=72
11:47:32.382 -> temperatureC=36.6250, IR=95103, INSTA_BPM=81.30, Avg BPM=72
11:47:32.382 -> temperatureC=36.6250, IR=95100, INSTA_BPM=81.30, Avg BPM=72
11:47:32.430 -> temperatureC=36.6250, IR=95106, INSTA_BPM=81.30, Avg BPM=72
11:47:32.430 -> temperatureC=36.6250, IR=95093, INSTA_BPM=81.30, Avg BPM=72
11:47:32.430 -> temperatureC=36.6250, IR=95110, INSTA_BPM=81.30, Avg BPM=72
11:47:32.477 -> temperatureC=36.6250, IR=95097, INSTA_BPM=81.30, Avg BPM=72
11:47:32.477 -> temperatureC=36.6250, IR=95096, INSTA_BPM=81.30, Avg BPM=72
11:47:32.524 -> temperatureC=36.6250, IR=95118, INSTA_BPM=81.30, Avg BPM=72
11:47:32.524 -> temperatureC=36.6250, IR=95129, INSTA_BPM=81.30, Avg BPM=72
11:47:32.571 -> temperatureC=36.6250, IR=95117, INSTA_BPM=81.30, Avg BPM=72
11:47:32.571 -> temperatureC=36.6250, IR=95136, INSTA_BPM=81.30, Avg BPM=72
11:47:32.571 -> temperatureC=36.6250, IR=95124, INSTA_BPM=81.30, Avg BPM=72
11:47:32.619 -> temperatureC=36.6250, IR=95139, INSTA_BPM=81.30, Avg BPM=72
11:47:32.619 -> temperatureC=36.6250, IR=95132, INSTA_BPM=81.30, Avg BPM=72
11:47:32.666 -> temperatureC=36.6250, IR=95144, INSTA_BPM=81.30, Avg BPM=72
11:47:32.666 -> temperatureC=36.6250, IR=95151, INSTA_BPM=69.77, Avg BPM=73
11:47:32.713 -> temperatureC=36.6250, IR=95167, INSTA_BPM=69.77, Avg BPM=73
11:47:32.713 -> temperatureC=36.6250, IR=95160, INSTA_BPM=69.77, Avg BPM=73
11:47:32.713 -> temperatureC=36.6250, IR=95169, INSTA_BPM=69.77, Avg BPM=73

```

**Figura 5.2 – Monitor seriale codice in figura 5.2**

Riassumendo, il seguente diagramma (figura 5.3a) mostra i vari step che l'algoritmo affronta per la misurazione della frequenza cardiaca.



**Figura 5.3a – Diagramma di flusso algoritmo frequenza cardiaca #1 (created with [draw.io](https://draw.io))**

### 5.a.ii – Codice

Figure 5.4a, 5.4b, 5.4c.

Il codice definitivo è disponibile al link [GitHub](#)

```

1  /*
2  Descrizione:
3  Codice per la lettura della frequenza cardiaca del sensore MAX30102
4  Autore: Caciolo Valerio
5  Data: 10/01/2022
6
7  Si ringrazia SparkFun per l'utilizzo della libreria "SparkFun_MAX3010x_Sensor_Library"
8  https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library
9  */
10
11 //librerie incluse
12 #include <Wire.h>
13 #include "MAX30105.h"
14 #include "heartRate.h"
15
16 //dichiarazioni e variabili globali
17 MAX30105 heartSensor;
18 const byte heartAvgArraySize = 15; //dimensione dell'array della media aritmetica dei battiti
19 byte heartAvgArray[heartAvgArraySize]; //array della media aritmetica dei battiti
20 byte indexArrayAvg = 0;
21 long exBeatsTime = 0;
22 long delta = 0; //variabile dove verra' salvato il tempo trascorso tra una battito e l'altro
23 float beatsPerMinute = 0;
24 float temperature;
25 int beatAvg = 0;
26
27 const int buzzer = 9;

```

Figura 5.4a - Rilevamento FC & Temp codice: dichiarazioni e librerie

```

30 void setup() {
31     //Init della porta seriale
32     Serial.begin(115200);
33     /*
34     Se gli occhiali non sono collegati al pc questa parte va lasciata commentata
35     while (!Serial) {
36         | ;
37     }
38     */
39     pinMode(buzzer, OUTPUT);
40
41     //Inizializzazione e controllo dell'hardware del MAX30102
42     if (!heartSensor.begin(Wire, I2C_SPEED_FAST)) //porta fast di default per I2C, 400kHz
43     {
44         Serial.println("MAX30105 non trovato. Problema hardware. Blocco sistema");
45         while (1);
46     }
47     Serial.println("Sensore MAX30102 correttamente riconosciuto");
48
49     heartSensor.setup(); //metodo per configurare il sensore con i valori di default del prodotto
50     heartSensor.setPulseAmplitudeRed(0x0A); //accensione led rosso
51     heartSensor.setPulseAmplitudeGreen(0); //spegnimento led verde
52     heartSensor.enableDIETEMPRDY(); // att temperatura
53 }

```

Figura 5.4b - Rilevamento FC & Temp codice: setup()

```

void loop() {
1   if(mytimer(2000))
2   {
3       temperature = heartSensor.readTemperature();
4   }
5
6   long irValue = heartSensor.getIR(); //valore ir direttamente dal Max30102
7
8   if (irValue >= 50000)
9   {
10      if (checkForBeat(irValue) == true)
11      {
12          //rilevato un battito riproduco un bip dal buzzer
13          tone(buzzer, 2000);
14          delay(10);
15          noTone(buzzer);
16
17          long tempTime = millis(); //salvo tempo corrente
18          delta = tempTime - exBeatsTime; // tempo trascorso tra battito corrente e precedente
19          exBeatsTime = tempTime;
20
21          beatsPerMinute = 60 / (delta / 1000.0);
22
23          if (beatsPerMinute < 255 && beatsPerMinute > 35)
24          {
25              heartAvgArray[indexArrayAvg++] = (byte)beatsPerMinute; //aggiungo la lettura nell'arra
26              indexArrayAvg %= heartAvgArraySize; //Wrap variable
27              //media delle letture
28              beatAvg = 0;
29              for (byte x = 0 ; x < heartAvgArraySize ; x++)
30                  beatAvg += heartAvgArray[x];
31              beatAvg /= heartAvgArraySize;
32          }
33          Serial.print("Body Temperature °C = ");
34          Serial.print(temperature+1.4 , 4);
35          Serial.print("IR=");
36          Serial.print(irValue);
37          Serial.print(", BPM=");
38          Serial.print(beatsPerMinute);
39          Serial.print(", Avg BPM=");
40          Serial.println(beatAvg);
41      }
42      }else
43      {
44          Serial.println("Occhiali non indossati!");
45          delay(2000);
46      }
47  }

```

**Figura 5.4c** - Rilevamento FC & Temp codice: loop()

## 5.b – Rilevamento delle cadute

Il paragrafo seguente presenta l'algoritmo che permette il rilevamento di cadute accidentali sfruttando i dati raccolti dal sensore LSM6DSOXTR IMU a 6 assi integrato nell'Arduino Nano 33 IoT.

### 5.b.i – L'algoritmo

L'algoritmo fa uso della libreria "Arduino\_LSM6DS3.h"<sup>20</sup> per leggere i dati dal giroscopio e dall'accelerometro del sensore LSM6D. Per il suo sviluppo si è preso in esame uno studio condotto dalla *Catholic University of America di Washington* e della *Vietnam Nation University di Ho Chi Minh City* [14].

I parametri utilizzati per il rilevamento sono i seguenti:

- il vettore della somma totale di accelerazione,

$$\vec{acc} = \sum_i \vec{acc}_i = \begin{pmatrix} a_x \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ a_y \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ a_z \end{pmatrix} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}; \quad |\vec{acc}| = \sqrt{(a_x)^2 + (a_y)^2 + (a_z)^2};$$

con  $a_x, a_y, a_z$  valori in  $g$  (o  $m^2/s$ ) rilevati dall'accelerometro rispettivamente sugli assi x, y, z;

- la velocità angolare totale,

$$\vec{\omega} = \sum_i \vec{\omega}_i = \begin{pmatrix} \omega_x \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \omega_y \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \omega_z \end{pmatrix} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}; \quad |\vec{\omega}| = \sqrt{(\omega_x)^2 + (\omega_y)^2 + (\omega_z)^2};$$

con  $\omega_x, \omega_y, \omega_z$  valori in  $^\circ/s$  rilevati dal giroscopio rispettivamente sugli assi x, y, z.

Quando gli occhiali sono fermi (in equilibrio),  $acc$  vale  $+1g$  e  $\omega$  vale  $0^\circ/s$ . In caso di caduta l'accelerazione cambia rapidamente e la velocità angolare produce una varietà di segnali lungo la traiettoria di caduta.

Perché la caduta sia rilevata vengono preliminarmente definite delle "soglie critiche" in tre step di calcolo:

- Nel primo step (Trigger1) si controlla se il valore attuale dell'accelerazione totale ( $acc$ ) superi il valore della soglia critica inferiore  $LFT_{acc}$ . Se questo avviene allora si controlla anche il superamento della soglia critica superiore  $UFT_{acc}$  (figura 5.5). Quando avviene una caduta, l'accelerazione subisce prima una piccola decrescita negativa dal suo valore di equilibrio, per poi subire un veloce aumento. Se entrambe le soglie del primo step sono superate, è possibile che ci troviamo in una situazione di potenziale caduta, quindi si passa al secondo step.

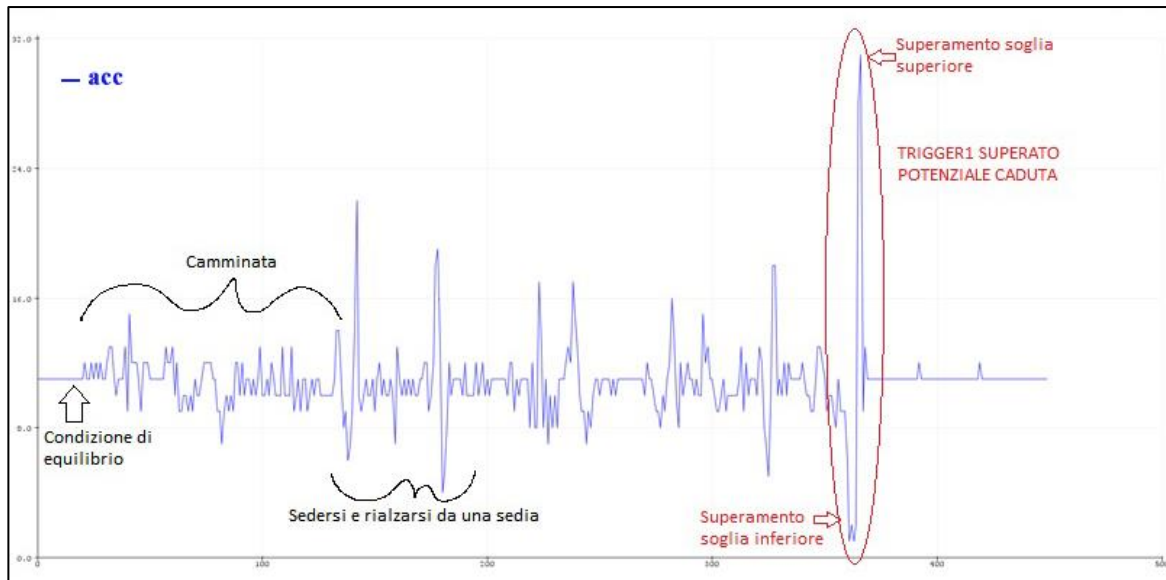


Figura 5.5 – Plotter seriale valore acc per trigger1

- nel secondo step (Trigger2) viene monitorata la variazione di angolazione attraverso la determinazione del valore della velocità angolare. Se la variazione di angolazione calcolata è tanto significativa da superare la soglia critica  $UFT_{gyro}$  (figura 5.6) ci sono buone probabilità che si stia verificando una caduta e si passa quindi all'ultimo step;

<sup>20</sup> Libreria ufficiale di Arduino per l'utilizzo del sensore LSM6DSOXTR dell'Arduino Nano 33 IoT. [Reference](#)

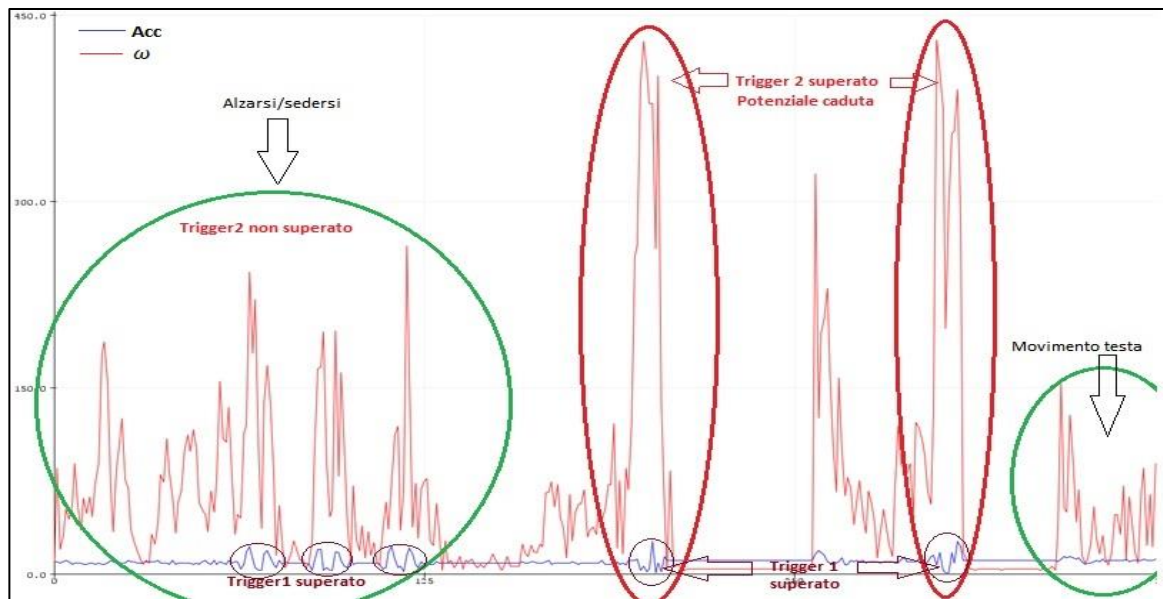


Figura 5.6 – Plotter seriale valore acc e valore  $\omega$  per trigger1 e trigger2

- nell'ultimo step (Trigger3) si assume preliminarmente che il soggetto stia cadendo e che stia per impattare il terreno; si cerca quindi di individuare il momento dell'impatto, ovvero il lasso di tempo delimitato in cui la velocità angolare ritorna ai valori di equilibrio. Se in questo lasso di tempo non viene registrato alcun movimento angolare possiamo desumere che la caduta si sia effettivamente verificata e che quelli rilevati negli step precedenti non fossero movimenti volontari.

Lo step 3 è fondamentale per distinguere due possibili esiti:

- se ci troviamo nella situazione in cui trigger1=true, trigger2=true e trigger3=false, viene inoltrata una notifica di possibile caduta/movimento avventato in cui si invita il ricevente ad accertarsi delle condizioni del soggetto osservato;
- se ci troviamo nella situazione in cui trigger1=true, trigger2=true e trigger3=true, viene inoltrata una notifica di emergenza di avvenuta caduta del soggetto.

Riassumendo, il seguente diagramma (figura 5.7) mostra i vari step dell'algoritmo

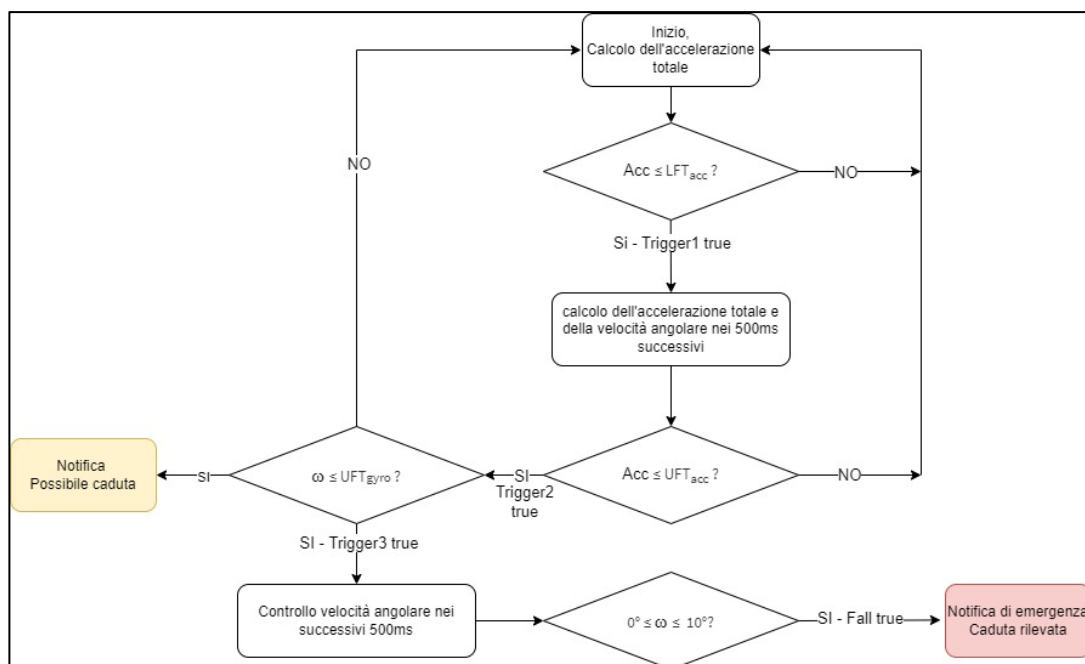


Figura 5.7 – Diagramma di flusso algoritmo cadute (created with [draw.io](https://draw.io))

I valori delle soglie  $UFT_{gyro}$ ,  $LFT_{acc}$  e di  $UFT_{acc}$  sono stati impostati seguendo lo studio preso in esame nel corso del capitolo. Nel capitolo seguente questi valori verranno modificati a seguito dell'analisi dei dati raccolti, permettendo così un migliore rilevamento delle cadute.



## 5.b.ii – Codice

Figure 5.8a, 5.8b, 5.8c, 5.8d. Non è presente la parte inerente all'inoltro degli avvisi di emergenza, verrà vista in seguito.

Il codice definitivo è disponibile al link [GitHub](#)

```

1  /*
2  Descrizione:
3  Codice per il rilevamento di una caduta
4  Autore: Caciolo Valerio
5  Data: 17/12/2021
6
7  Si ringrazia gli autori dello studio "Optimization of an Accelerometer and Gyroscope-Based
8  Fall Detection Algorithm"
9  https://www.hindawi.com/journals/js/2015/452078/
10 */
11
12 //librerie include
13 #include <Arduino_LSM6DS3.h>
14
15 //dichiarazioni e variabili globali
16 const int buzzer = 9;
17 float ax = 0, ay = 0, az = 0, gx = 0, gy = 0, gz = 0;
18
19 boolean fall = false; //true se si è rilevata una caduta
20 boolean trigger1=false; //trigger per la soglia critica inferiore dell'accelerazione totale
21 boolean trigger2=false; //trigger per la soglia critica superiore dell'accelerazione totale
22 boolean trigger3=false; //trigger per la soglia critica superiore della velocità angolare
23 /*
24 i seguenti contatori servono ad impostare un tempo a partire dal quale ogni trigger attende all'incirca
25 500ms che si verifichi il trigger successivo
26 */
27 byte trigger1count = 0; //conta i cicli in cui trigger1 = true
28 byte trigger2count = 0; //conta i cicli in cui trigger2 = true
29 byte trigger3count = 0; //conta i cicli in cui trigger3 = true
30
31 int omega = 0;
32 float acc = 0;

```

Figura 5.8a – Rilevamento cadute codice: dichiarazioni e librerie

```

34 void setup(){
35     Serial.begin(9600);
36     //while (!Serial);
37     pinMode(buzzer, OUTPUT);
38
39     //Inizializzazione e controllo dell'hardware del LSM6DS3
40     if (!IMU.begin())
41     {
42         Serial.println("Failed to initialize IMU!");
43         while (1);
44     }
45
46     Serial.print("Gyroscope sample rate = ");
47     Serial.print(IMU.gyroscopeSampleRate());
48     Serial.println(" Hz");
49     Serial.println();
50     Serial.println("Gyroscope in degrees/second");
51
52     Serial.print("Accelerometer sample rate = ");
53     Serial.print(IMU.accelerationSampleRate());
54     Serial.println(" Hz");
55     Serial.println();
56     Serial.println("Acceleration in G's");
57     Serial.println("X\tY\tZ");
58 }

```

Figura 5.8b - Rilevamento cadute codice: setup()

```

138 void leggiLSM6DS3(){
139     IMU.readGyroscope(gx, gy, gz);
140     IMU.readAcceleration(ax, ay, az);
141 }

```

Figura 5.8c - Rilevamento cadute codice: funzione leggiLSM6DS3()



```

60 void loop(){
61   if(IMU.accelerationAvailable() && IMU.gyroscopeAvailable())
62   {
63     leggiLSM6DS3();
64   }
65   acc = pow(pow(ax,2)+pow(ay,2)+pow(az,2),0.5);
66   int accMod = acc * 10; //visto che i valori di equilibrio sono da 0 a 1
67   Serial.println(accMod);
68
69   if (trigger3==true)
70   {
71     omega = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5);
72     trigger3count++;
73     //Serial.println(trigger3count);
74     if (trigger3count>=10)
75     {
76       omega = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5);
77       Serial.println(omega);
78       if ((omega>=0) && (omega<=10))
79       { //ritorno a situazione di equilibrio quindi impatto
80         fall=true; trigger3=false; trigger3count=0;
81         Serial.println(omega);
82       }
83       else
84       { //probabilmente era un movimento
85         trigger3=false; trigger3count=0;
86         Serial.println("TRIGGER 3 DEACTIVATED");
87       }
88     }
89   }
90   if (fall==true)
91   { //se è stata rilevata una caduta faccio suonare il buzzer
92     Serial.println("FALL DETECTED");
93     tone(buzzer, 1000);
94     delay(5000);
95     noTone(buzzer);
96     fall=false;
97     // exit(1);
98   }
99   if (trigger2count>=6)
100   { //nei prossimi 0.5 secondi si cercherà il superamento della soglia critica superiore di omega
101     trigger2=false; trigger2count=0;
102     Serial.println("TRIGGER 2 DEACTIVATED");
103   }
104   if (trigger1count>=6)
105   { //nei prossimi 0.5 secondi si cercherà il superamento della soglia critica superiore di accMod
106     trigger1=false; trigger1count=0;
107     Serial.println("TRIGGER 1 DEACTIVATED");
108   }
109   if (trigger2==true)
110   {
111     trigger2count++;
112     omega = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5); Serial.println(omega);
113     if (omega>=30 && omega<=400)
114     { //SOGLIA CRITICA SUPERIORE OMEGA
115       trigger3=true; trigger2=false; trigger2count=0;
116       Serial.println(omega);
117       Serial.println("TRIGGER 3 ACTIVATED");
118     }
119   }
120   if (trigger1==true)
121   {
122     trigger1count++;
123     if (accMod>=11)
124     { //SOGLIA CRITICA SUPERIORE ACCMOD
125       trigger2=true;
126       Serial.println("TRIGGER 2 ACTIVATED");
127       trigger1=false; trigger1count=0;
128     }
129   }
130   if (omega<=3 && trigger2==false)
131   { //SOGLIA CRITICA INFERIORE ACCMOD
132     trigger1=true;
133     Serial.println("TRIGGER 1 ACTIVATED");
134   }
135   delay(100);
136 }

```

Figura 5.8d - Rilevamento cadute codice: loop()

## 5.c – Integrazione con Arduino IoT Cloud

Come illustrato nei precedenti capitoli, le due funzioni principe degli “SE Eyeglasses” consistono nella possibilità di monitorare i parametri vitali del soggetto e nella capacità di inoltrare notifiche di allerta in risposta al presentarsi di situazioni di pericolo per l’utente, nello specifico il rilevamento di una caduta accidentale e la presenza di valori anomali nella frequenza cardiaca o/e nella temperatura corporea.

Per il prototipo degli occhiali si è scelto di utilizzare la piattaforma cloud “Arduino IoT Cloud”.

### 5.c.i – Features

La piattaforma cloud messa a disposizione da Arduino consente, tramite un’apposita interfaccia grafica (figura 5.9), di controllare in tempo reale i parametri di frequenza cardiaca e temperatura corporea con un relativo grafico di andamento. È inoltre in grado, attraverso dei riquadri di status, di mostrare in tempo reale se gli occhiali sono indossati, se è avvenuta una caduta e di inoltrare notifiche di allerta.



Figura 5.9 – Arduino IoT Cloud, interfaccia grafica

### 5.c.ii – Connessione con gli occhiali

La comunicazione tra gli occhiali ed il cloud avviene con l’uso del modulo WiFi NINA-W10 integrato nella scheda Arduino Nano 33 IoT e delle librerie *ArduinoIoTCloud.h*<sup>21</sup> e *Arduino\_ConnectionHandler.h*<sup>22</sup>.

Nella fase iniziale di avvio Arduino cerca costantemente di connettersi alla rete WiFi definita nel file “*Secret.h*”; a quel punto stabilisce una connessione con il cloud attraverso la funzione “*ArduinoCloud.begin(ArduinoIoTPreferredConnection)*”. Ogni due secondi la funzione “*Arduino.update()*” esegue un upload dei dati inerenti la frequenza cardiaca e la temperatura corporea, mentre in caso di caduta l’upload viene effettuato istantaneamente. Esso avviene su delle variabili specifiche definite in fase di preparazione del cloud: *bool caduta*; *bool possibileCaduta*; *bool wearGlasses*; *float currentBpm*; *float bodyTemperature*;

<sup>21</sup> Libreria ufficiale di Arduino per le funzioni di Arduino IoT Cloud per la scheda Arduino Nano 33 IoT. [Reference](#)

<sup>22</sup> Libreria ufficiale di Arduino per gestire la connessione tra Arduino IoT Cloud e la scheda Arduino Nano 33 IoT. [Reference](#)

## Capitolo 6.

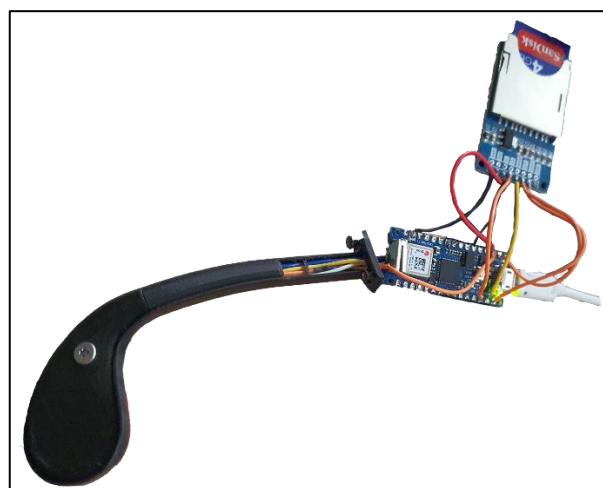
# Raccolta ed analisi dei dati

In questo capitolo sono presentate tutte le riflessioni, le modifiche e le considerazioni effettuate a seguito della raccolta e dello studio dei dati registrati dagli occhiali, con particolare focus sulla frequenza cardiaca e sul rilevamento delle cadute.

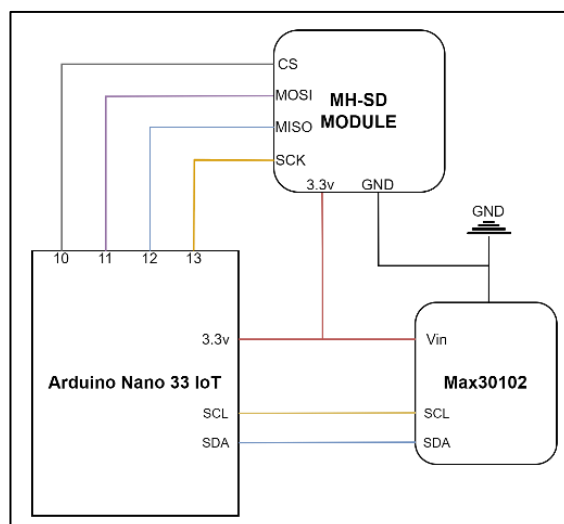
### 6.a – Metodologia

L'acquisizione costante dei dati, soprattutto alla velocità con cui gli stessi venivano prodotti, è stata una delle sfide più impegnative nella realizzazione del prototipo. Per la fase di raccolta dei dati sono stati creati due appositi script: *"fcLogger.ino"* e *"fallLogger.ino"*. Con ciascuno degli script il microcontrollore è in grado di generare circa cinquanta stringhe al secondo con i dati di interesse, il che equivale ad una lettura circa ogni 1.2 ms. E' facile dedurre come effettuare una raccolta tramite Cloud con il modulo WiFi del microcontrollore sia stato perciò piuttosto ostico: vista l'esigua memoria del microcontrollore, anche provando a collezionare i dati in apposite strutture dati non risultava possibile inoltrarli in modo efficiente.

Si è concluso che l'unico modo per effettuare un log dei dati fosse quello di implementare un MH-SD Card Module (*figura 6.1*) che consentisse di ampliare la memoria di massa dell'Arduino permettendo, attraverso la libreria *"SD.h"*, di creare, scrivere e leggere file *.txt*.



**Figura 6.1** – Arduino con modulo SDcardReader



**Figura 6.2** - Circuit Design SD module (created with [draw.io](https://draw.io))

Entrambi i codici utilizzati per effettuare i log della frequenza cardiaca e dei valori per il rilevamento delle cadute sono disponibili rispettivamente ai seguente link: [GitHubFcLogger](#); [GitHubFallLogger](#).

Per quanto riguarda lo studio dei dati prodotti dal pulsossimetro MAX30102, si è deciso di analizzare la qualità della misurazione per affinare l'algoritmo che identifica la frequenza cardiaca. Sono state raccolte diverse tranches di dati: nella prima gli occhiali sono stati indossati dal soggetto in un momento di relax, seduto ad una scrivania o davanti al computer; nella seconda e nella terza il soggetto ha svolto un'intensa attività fisica alternata a momenti di pausa, per determinare la velocità di assestamento e le oscillazioni nella

rilevazione della corretta frequenza cardiaca (utile ai fini del monitoraggio accurato e puntuale per l'eventuale inoltro di notifiche di allerta in caso di valori anomali dei bpm).

Nella quarta e quinta tranche il soggetto ha svolto leggere attività fisiche cercando di sollecitare a più riprese il sensore, eseguendo movimenti rapidi della testa. Queste ultime misurazioni sono state eseguite successivamente all'apporto di miglioramenti all'algoritmo per verificare l'efficacia degli stessi.

Relativamente al rilevamento delle cadute si è invece cercato di definire in maniera più accurata possibile i valori associati alle soglie critiche  $UFT_{gyro}$ ,  $LFT_{acc}$  e di  $UFT_{acc}$  definite nel capitolo precedente. A tal fine ci si è serviti delle seguenti formule per determinare la *sensibilità*, la *precisione* e la *specificità*:

$$- Sensitivity = \frac{\#TP}{\#TP + \#FN}; \quad - Accuracy = \frac{\#TN + \#TP}{\#TN + \#FN + \#TP + \#FP}; \quad - Specificity = \frac{\#TN}{\#TN + \#FP};$$

#TP (*True positive*): la caduta si è verificata e il dispositivo l'ha rilevata

#FN (*False negative*): la caduta si è verificata e il dispositivo non l'ha rilevata

#FP (*False positive*): la caduta non si è verificata ma il dispositivo l'ha rilevata

#TN (*True negative*): la caduta non si è verificata e il dispositivo non l'ha rilevata

Questi valori sono stati un buono strumento di valutazione della qualità generale delle rilevazioni. La difficoltà è consistita nel definire i valori ottimali per le soglie critiche affinché le formule di *sensibilità* e *specificità* fossero il più possibile bilanciate e tendenti al valore 1: una diminuzione dei valori delle soglie critiche superiori  $UFT_{gyro}$ ,  $UFT_{acc}$  e un aumento della soglia critica inferiore  $LFT_{acc}$  portava infatti ad un aumento di falsi positivi (#FP), determinando una decrescita del valore della *specificità* ma aumentando quello della *sensibilità*; viceversa, un aumento dei valori delle soglie critiche superiori  $UFT_{gyro}$ ,  $UFT_{acc}$  e una diminuzione della soglia critica inferiore  $LFT_{acc}$  comportava un aumento di falsi negativi (#FN), a scapito della *sensibilità* e un aumento della *specificità* con il rischio però di non segnalare un'effettiva caduta.

## 6.b – Risultati e confronto con prodotti simili

Tutte le seguenti rilevazioni e raccolte di dati sono state eseguite sui seguenti soggetti:

- uomo di 25 anni in salute, alto 1,78m e di 66kg di peso;
- uomo di 25 anni in salute, alto 1,81m e di 63kg di peso;
- donna di 63 anni in salute, alta 1,64m e di 59kg di peso;
- donna di 26 anni in salute, alta 1,72m e di 62kg di peso.

### 6.b.i – Rilevamento della frequenza cardiaca

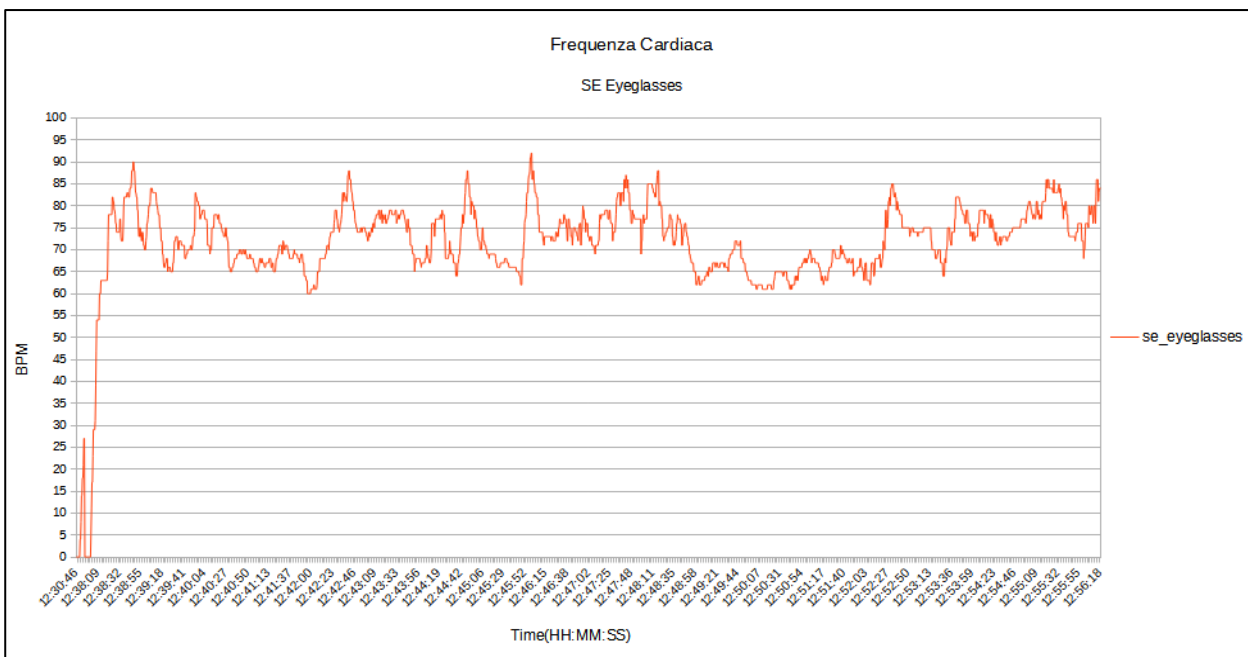
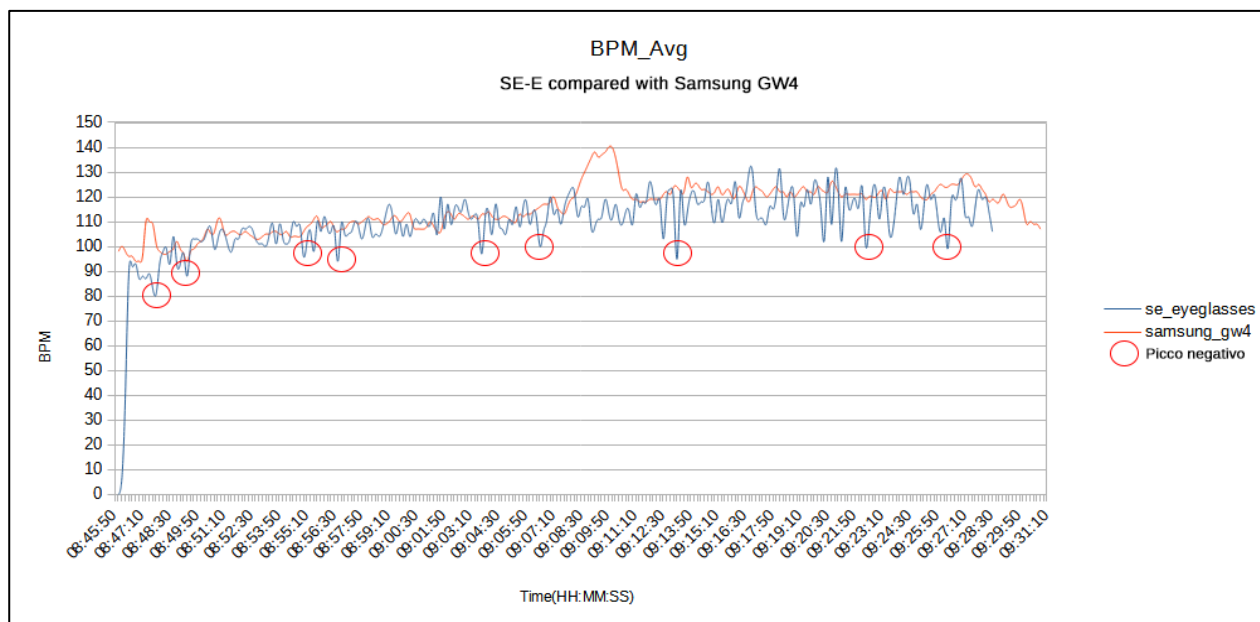


Figura 6.3 – Grafico misurazione frequenza cardiaca a riposo

Come precedentemente accennato, la prima raccolta dati è stata effettuata in uno stato di riposo del soggetto che alternava momenti di relax (in cui era seduto davanti al computer) a momenti di lieve attività fisica. Gli occhiali sono stati indossati per circa venti minuti a partire dalle ore 12:38 fino alle ore 12:56. Come si può evincere dal grafico riportato in figura 6.3, la misurazione ottenuta dagli occhiali, confrontata con quella prodotta da uno smartwatch Samsung, è stata abbastanza accurata: la frequenza cardiaca staziona nella fascia 60-75 bpm mentre il soggetto è seduto, raggiungendo invece la fascia 75-90 bpm durante le leggere attività fisiche.



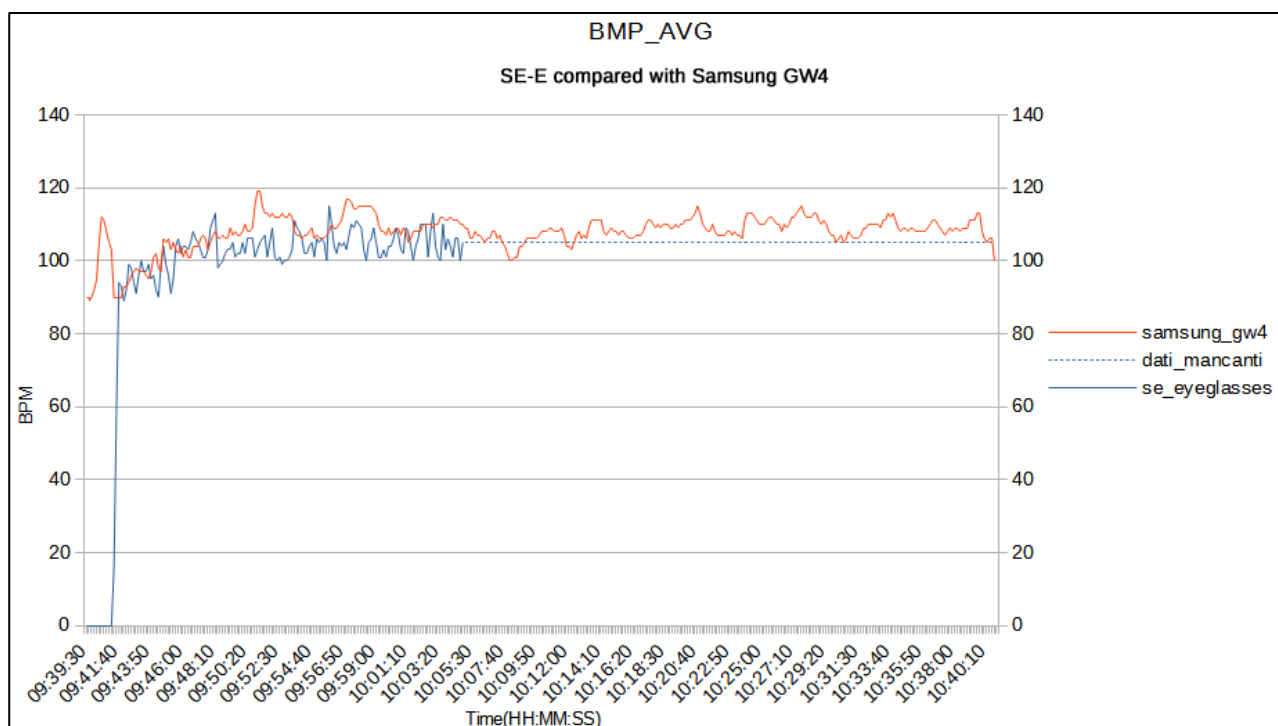
**Figura 6.4** – Grafico misurazione frequenza cardiaca sotto stress confrontata con GW4

Nel grafico in figura 6.4 è stata registrata una prima attività fisica intensa del soggetto. Anche in questo caso i dati sono stati raffrontati con quelli raccolti da un dispositivo wearable, quale uno smartwatch Samsung (riportati nel grafico in **arancione**). Dalla figura 6.4 emerge come in questa situazione si siano presentati i primi problemi relativi ai dati prodotti dagli “SE Eyeglasses”:

1. La frequenza cardiaca è leggermente sottostimata rispetto allo smartwatch, il che non presenta un grave problema;
2. La misurazione della frequenza cardiaca non è omogenea rispetto a quella misurata dallo smartwatch;
3. Ci sono dei picchi negativi (evidenziati in figura con un cerchio rosso **○**) che rappresentano un’errata misurazione da parte del sensore degli occhiali.

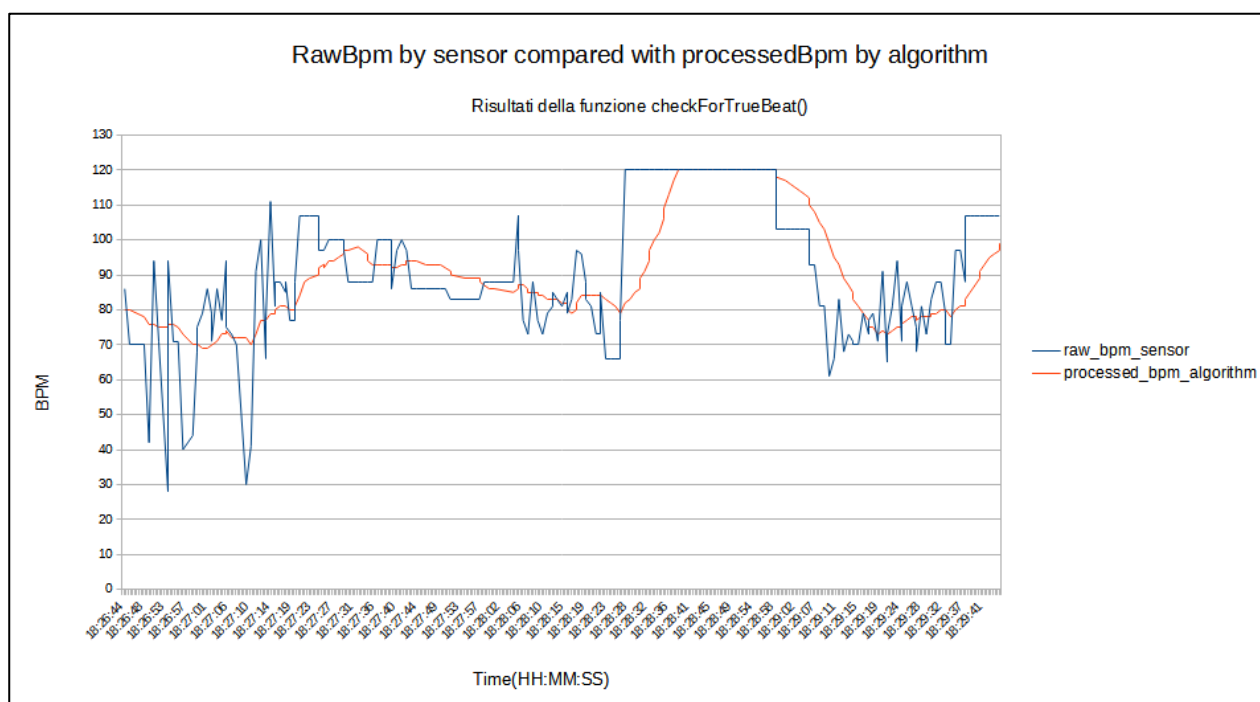
La totalità delle problematiche è ascrivibile alle seguenti concause:

- il sensore Max30102 non è di buona qualità. Può basare le sue misurazioni sull’illuminazione di un singolo led rosso (a bassa intensità) e sulla lettura di un singolo fotorelevatore. Al contrario, il Samsung Galaxy Watch possiede diversi led verdi ad alta intensità, sei fotorelevatori e vari elettrodi;
- ogni misurazione viene accettata senza tener conto di quella precedente. Se in una situazione di riposo del soggetto (come nella figura 6.3) la maggior parte delle misurazioni risulta corretta poiché il sensore mantiene una posizione pressoché costante, senza essere sottoposto a brusche sollecitazioni da parte dell’utente, in caso di attività intense e di sudorazione la maggiore sollecitazione causa misurazioni errate. Per risolvere questo problema si è implementata una semplice funzione chiamata “*bool checkForTrueBeat()*” che confronta il valore corrente misurato dal sensore con quello precedentemente registrato: se esso è accettabile (entro margini stabiliti a seguito di diversi collaudi) viene considerato, altrimenti viene corretto o scartato;
- ogni misurazione incide eccessivamente sulla media aritmetica eseguita nella coda, il che comporta rapide diminuzioni quando il sensore rileva una serie di valori errati. In risposta alla problematica è stata aumentata la capienza della coda a quindici elementi per permettere solo a un cospicuo numero di misurazioni simili di riuscire ad incidere significativamente sulla media aritmetica.



**Figura 6.5** – Grafico misurazione frequenza cardiaca sotto stress confrontata con GW4 dopo l'aumento di elementi nella coda

Come illustrato nel grafico riportato in figura 6.5, il solo aumento degli elementi della coda ha migliorato sensibilmente la situazione relativamente all'allineamento delle misurazioni registrate dagli occhiali e dallo smartwatch, riducendo notevolmente i picchi negativi dal precedente esempio.



**Figura 6.6** – Grafico di confronto tra i dati bpm grezzi prodotti dal sensore e quelli elaborati dall'algoritmo con `checkForTrueBeat()`

Il grafico in figura 6.6 riporta un confronto tra il dato grezzo prodotto dal sensore e quello prodotto dall'algoritmo a seguito dei vari step sopra meglio definiti (media aritmetica tra più misurazioni e il controllo con la funzione `checkForTrueBeat()`). Si può notare come le misurazioni siano decisamente più omogenee tra loro e, nonostante la permanenza di picchi negativi nel dato grezzo (relativi alle misurazioni errate), essi non influiscano sul valore finale post elaborazione dell'algoritmo grazie alla funzione `checkForTrueBeat()` che ne imposta il valore sulla base di quello precedentemente misurato.



Da uno studio approfondito del grafico in figura 6.6 è possibile osservare la sussistenza di un problema evidente della funzione *checkForTrueBeat()* intorno al tempo 18:28:28, scatenato da diversi fattori:

- il sensore ha prodotto una misurazione errata di 120bpm al tempo 18:28:28 (evidenziata in **rosso**) rispetto alla precedente, di valore 77, al tempo 18:28:27 (evidenziata in **verde**);
- visto che la misurazione della frequenza cardiaca presenta problemi solo al raggiungimento di valori superiori di 80/90bpm, l'algoritmo entra in azione solo quando la frequenza elaborata dall'algoritmo risulta essere maggiore o uguale a 80bpm. Sfortunatamente questa soglia è stata raggiunta (valore evidenziato in **giallo**) proprio nel momento di arrivo della misurazione errata che quindi è stata considerata comunque valida;
- essendo stata considerata valida, tutte le misurazioni corrette che si sono susseguite dopo il tempo 18:28:29 con valore uguale all'incirca a 70/80bpm sono state ritenute errate rispetto alla precedente 120bpm generando una serie di errori a cascata.
- La situazione si è poi risolta al tempo 18:29:06 in quanto la frequenza del soggetto si è alzata portando il sensore a fornire valori simili a 120bpm che quindi hanno ristabilito il funzionamento dell'algoritmo.

18:28:27	66	79
18:28:27	77	79
18:28:28	120	82
18:28:29	120	83
18:28:30	120	85
18:28:31	120	86
18:28:31	120	89
18:28:32	120	91
18:28:33	120	94
18:28:33	120	97
18:28:34	120	100
18:28:35	120	102
18:28:36	120	106
18:28:36	120	109
18:28:37	120	113
18:28:38	120	117
18:28:39	120	120
18:28:39	120	120
18:28:40	120	120
18:28:41	120	120
18:28:42	120	120
18:28:43	120	120
18:28:44	120	120
18:28:45	120	120
18:28:45	120	120
18:28:46	120	120
18:28:47	120	120
18:28:48	120	120
18:28:49	120	120
18:28:49	120	120
18:28:51	120	120
18:28:52	120	120
18:28:53	120	120
18:28:54	120	120
18:28:55	120	120
18:28:55	120	120
18:28:56	120	120
18:28:57	120	120
18:28:58	120	120
18:28:59	120	120
18:28:59	103	118
18:29:01	103	117
18:29:02	103	116
18:29:03	103	115
18:29:04	103	114

**Figura 6.7** – log file del grafico in figura 6.6

Per risolvere il problema sono stati aggiunti ulteriori controlli nella funzione che consentono di confrontare il valore grezzo sia con quello elaborato dall'algoritmo che con quelli presenti nella coda di quindici elementi. Se il valore risulta essere errato, viene scartato.

Di seguito, in figura 6.8, il codice della funzione *checkForTrueBeat()* a seguito delle modifiche e delle implementazioni apportate:

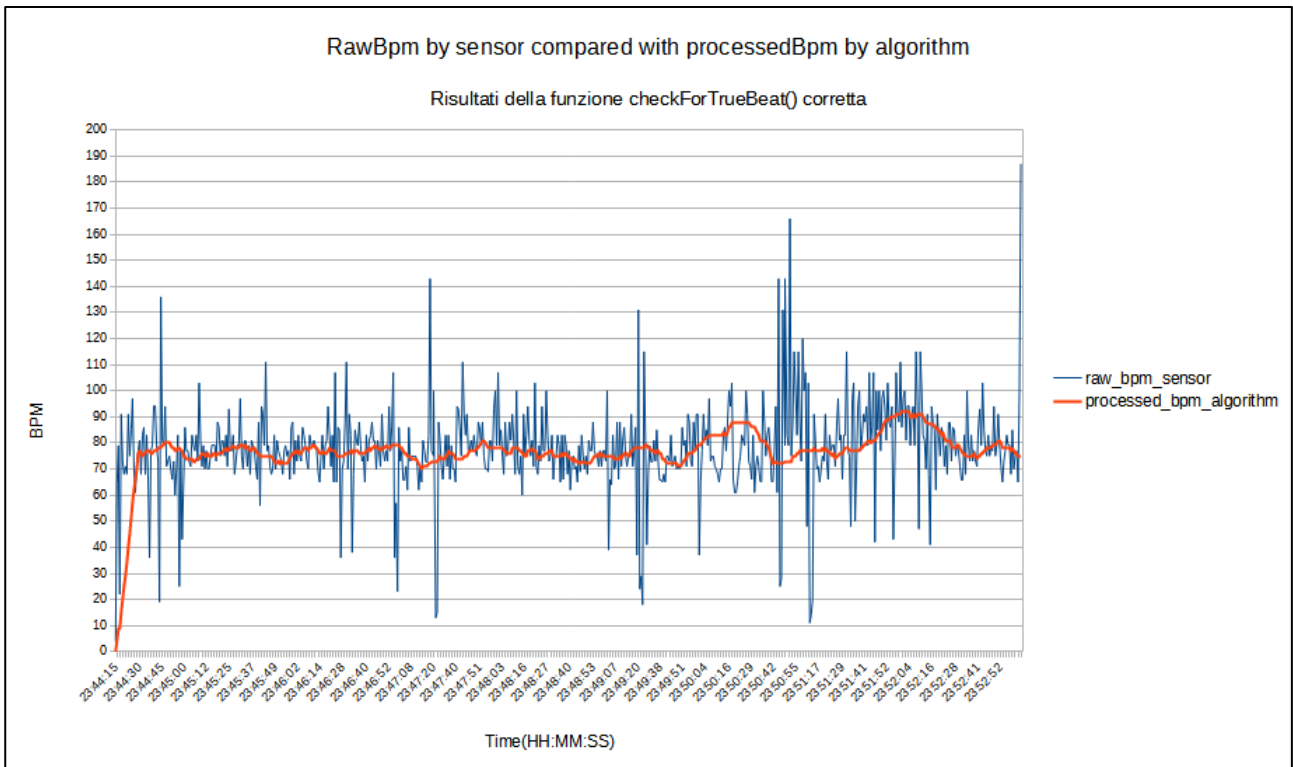
```

139 bool checkForTrueBeat(){
140     int suspiciousHeartRateCount = 0;
141     if (beatsPerMinute >= 200 && beatsPerMinute <= 35) return false;
142     if ((beatsPerMinute <= exBeatsPerMinute + 20 && beatsPerMinute >= exBeatsPerMinute - 20)
143         && (beatsPerMinute <= beatAvg + 20 && beatAvg >= exBeatsPerMinute - 20))
144     {
145         for (byte c = 0 ; c < heartAvgArraySize ; c++)
146         {
147             if(beatsPerMinute >= heartAvgArray[c] + 20 && beatsPerMinute <= heartAvgArray[c] - 20) suspiciousHeartRateCount++;
148         }
149         if(suspiciousHeartRateCount >= 10) return false;
150         return true;
151     }
152     return false;
153 }

```

**Figura 6.8** – Rilevamento FC & Temp codice: funzione *checkForTrueBeat()*

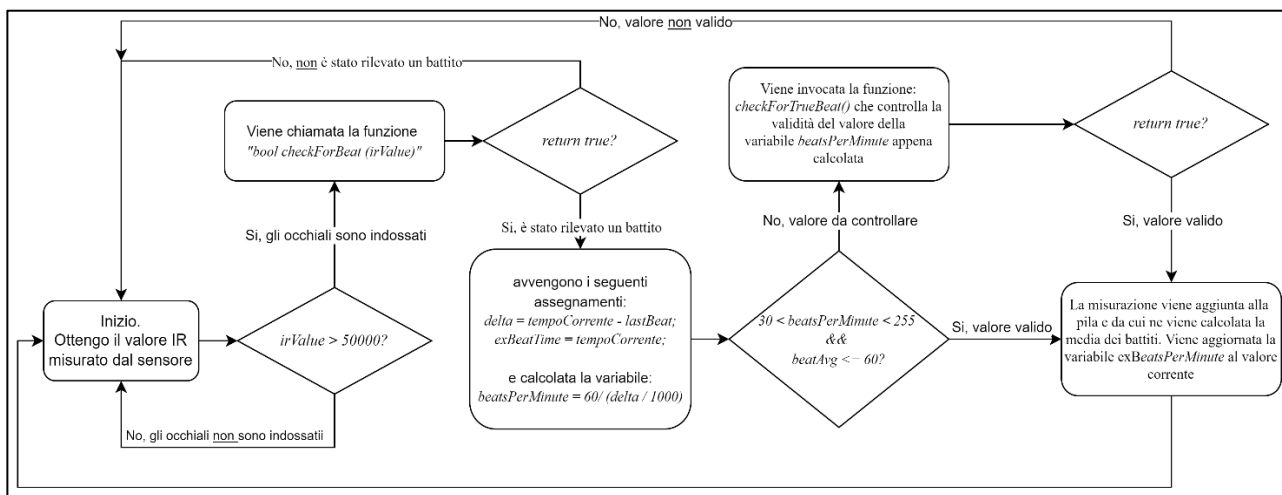
Ecco il risultato ottenuto successivamente all'implementazione dell'ultima versione del codice (figura 6.9):



**Figura 6.9** – Confronto tra i dati bpm grezzi prodotti dal sensore e quelli elaborati dall'algoritmo con `checkForTrueBeat()` migliorata

Il dato processato dall'algoritmo risulta omogeneo nonostante le numerose misurazioni errate registrate dal sensore. Inoltre, confrontando il risultato ottenuto con la misurazione effettuata contemporaneamente dallo smartwatch, le due serie di dati sono praticamente sovrapponibili, il che denota una buona capacità degli occhiali di adattarsi a variazioni continue della frequenza cardiaca.

Segue il diagramma di flusso (figura 5.3b) aggiornato a seguito delle migliorie apportate all'algoritmo:



**Figura 5.3b** - Diagramma di flusso algoritmo frequenza cardiaca #2 (created with [draw.io](https://draw.io))

L'intero codice (completo e definitivo) per il calcolo della frequenza cardiaca e della temperatura corporea è disponibile al link [GitHub](https://github.com).



### 6.b.ii – Rilevamento delle cadute

Per l'esecuzione dei test il circuito è stato leggermente modificato (figura 6.10) aggiungendo un piccolo buzzer in grado di riprodurre semplici ma potenti suoni a 8 bit in risposta alla rilevazione di una caduta da parte dell'hardware. Per evitare danni all'hardware stesso degli occhiali, durante i test è stata prodotta una nuova struttura (figura 6.11), più semplice e resistente, ma comunque in grado di simulare un paio di occhiali.

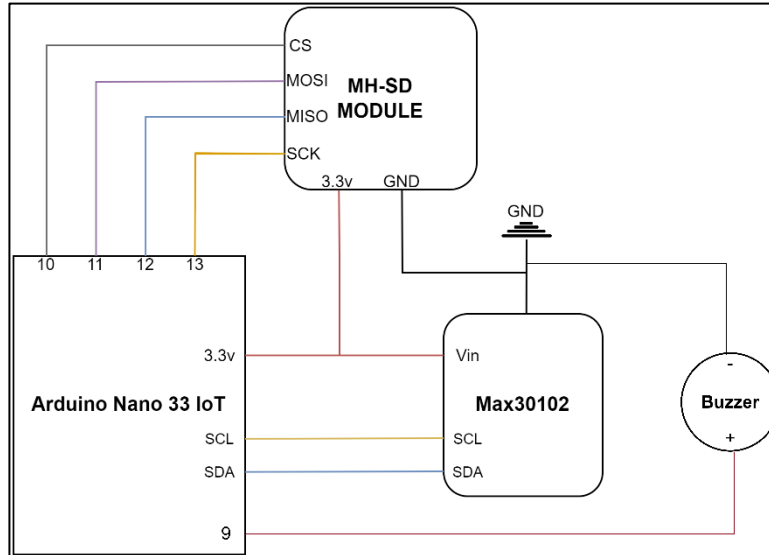


Figura 6.10 - Circuit Design SD module with buzzer (created with [draw.io](https://draw.io))

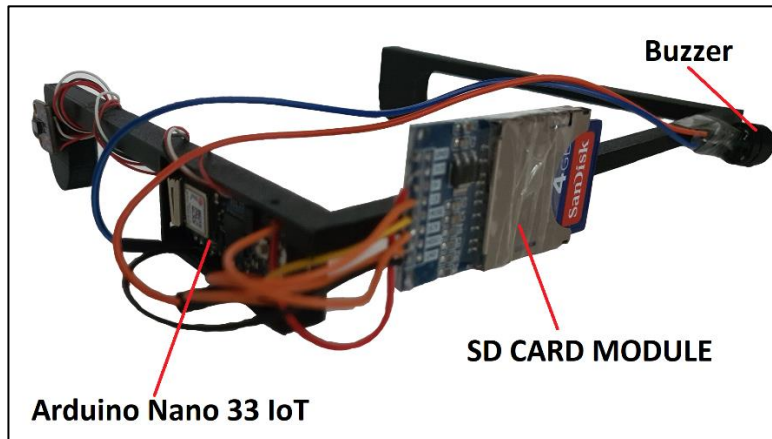


Figura 6.11 – Struttura per l'analisi dati del rilevamento delle cadute

Serie 1: (figure 6.12, 6.13, 6.14)

I valori delle soglie critiche durante la prima serie di test sono stati i seguenti:

- $LFT_{acc} = 3g$ , quindi  $trigger1 = true \text{ sse } acc \leq 3g$
- $UFT_{acc} = 11g$ , quindi  $trigger2 = true \text{ sse } acc \geq 11g$
- $30^\circ/s \leq UFT_{gyro} \leq 400^\circ/s$ , quindi  $trigger3 = true \text{ sse } 30^\circ/s \leq \omega \leq 400^\circ/s$

Con essi è stato eseguito il primo test, di circa un'ora, durante il quale si è cercato di simulare un utilizzo standard del dispositivo, svolgendo azioni quotidiane oltre a cadute simulate e annotando su un diario le varie azioni svolte e il relativo conteggio del valore delle variabili per il calcolo delle formule di *sensibilità*, *precisione* e *specificità*. Sono stati registrati i seguenti valori su 30 test, comprendenti 10 cadute simulate:

#TP (True positive) = 10  
#FN (False negative) = 0

#FP (False positive) = 11  
#TN (True negative) = 9

A seguito dei risultati i valori delle tre formule sono stati i seguenti:

$$Sensitivity = \frac{\#TP}{\#TP + \#FN} = \frac{10}{10 + 0} = 1$$

$$Specificity = \frac{\#TN}{\#TN + \#FP} = \frac{11}{11 + 9} = 0.55$$

$$Accuracy = \frac{\#TN + \#TP}{\#TN + \#FN + \#TP + \#FP} = \frac{9 + 10}{9 + 0 + 10 + 11} = 0.63$$

Con i valori scelti per le soglie critiche in questa prima serie di test l'algoritmo ha raggiunto un ottimo risultato di 1/1 relativamente alla *sensibilità*: tutte le cadute verificatesi sono state infatti correttamente rilevate. È stato però registrato un alto numero di falsi positivi dovuto alla *specificità*, con valore di 0.55/1. Nonostante l'elevata sensibilità, una configurazione simile presenta il 45% di probabilità di segnalare false cadute, percentuale troppo alta per essere considerata accettabile.

Segue un'analisi dei dati registrati nel *log.txt*, suddivisi in sezioni definite a seconda dei movimenti eseguiti e annotati sul diario.

- Veloci movimenti della testa (figura 6.12).

in esame un lasso di tempo dalle 18:18:59 alle 18:19:10 durante il quale sono stato eseguiti rapidi movimenti della testa finalizzati ad ingannare l'algoritmo. I dati evidenziati in verde rappresentano i valori che superano la soglia critica  $UFT_{acc}$

Time(HH:MM:SS)	acc	$\omega$	Time(HH:MM:SS)	acc	$\omega$	Time(HH:MM:SS)	acc	$\omega$
18:18:59	9	7	18:19:03	9	7	18:19:07	9	7
18:18:59	9	7	18:19:03	9	7	18:19:07	9	7
18:18:59	9	7	18:19:03	9	7	18:19:07	9	7
18:18:59	9	7	18:19:03	9	7	18:19:07	9	7
18:18:59	9	7	18:19:03	10	7	18:19:07	9	7
18:18:59	10	7	18:19:03	9	7	18:19:07	9	7
18:19:00	11	7	18:19:03	9	7	18:19:08	9	7
18:19:00	9	7	18:19:04	9	7	18:19:08	9	7
18:19:00	9	7	18:19:04	9	7	18:19:08	9	7
18:19:00	10	7	18:19:04	9	7	18:19:08	9	7
18:19:00	9	7	18:19:04	9	7	18:19:08	9	7
18:19:00	9	7	18:19:04	9	7	18:19:08	9	7
18:19:00	9	7	18:19:04	9	7	18:19:08	9	7
18:19:01	10	7	18:19:04	9	7	18:19:09	9	7
18:19:01	9	7	18:19:05	9	7	18:19:09	11	7
18:19:01	9	7	18:19:05	9	7	18:19:09	9	7
18:19:01	9	7	18:19:05	9	7	18:19:09	9	7
18:19:01	10	7	18:19:05	9	7	18:19:09	9	7
18:19:01	9	7	18:19:05	9	7	18:19:09	9	7
18:19:01	10	7	18:19:05	9	7	18:19:09	9	7
18:19:02	10	7	18:19:06	9	7	18:19:10	10	7
18:19:02	9	7	18:19:06	9	7	18:19:10	8	7
18:19:02	9	7	18:19:06	9	7	18:19:10	8	7
18:19:02	9	7	18:19:06	9	7	18:19:10	10	7
18:19:02	9	7	18:19:06	9	7	18:19:10	15	7
18:19:02	10	7	18:19:06	10	7	18:19:10	14	7
18:19:02	9	7	18:19:06	9	7	18:19:10	10	7

Figura 6.12 – Serie 1, esempio 1.

Nonostante la soglia  $UFT_{acc}$  venga superata più volte (dati evidenziati in verde), l'algoritmo si comporta in maniera soddisfacente, senza segnalare falsi positivi. Questo tipo di comportamento è dovuto al fatto che, perché la misurazione sia considerata accettabile, è necessario soddisfare il primo trigger dell'algoritmo che controlla la condizione della soglia critica inferiore  $LFT_{acc} \leq 3g$ .

Inoltre, si può notare come il valore di  $\omega$  resti costante in quanto la velocità angolare viene calcolata solo in caso di superamento del secondo trigger.

▪ Cadute simulate (figura 6.13).

In esame 3 delle 10 cadute, avvenute rispettivamente alle ore 18:18:38, 18:22:24 e 18:22:59.

Ogni Trigger che rappresenta il superamento di una delle soglie critiche è evidenziato in verde insieme al valore che ne ha causato l'attivazione.

Prima di analizzare il log di questa prova è doveroso fare una premessa: una caduta simulata non è paragonabile ad una caduta effettiva. Il valore dell'accelerazione della prima è infatti da considerare sempre più contenuto rispetto a quello della seconda, anche perché durante le simulazioni i soggetti dei test si sono gettati su un materasso posizionato a terra ma con non troppa violenza.

Caduta 1			Caduta 2			Caduta 3 (più blanda)		
Time(HH:MM:SS)	acc	$\omega$	Time(HH:MM:SS)	acc	$\omega$	Time(HH:MM:SS)	acc	$\omega$
18:18:38	10	7	18:22:23	8	7	18:22:59	10	6
18:18:38	13	7	18:22:24	8	7	18:22:59	6	6
18:18:38	8	7	18:22:24	5	7	18:22:59	3	6
18:18:38	2	7	18:22:24	2	7	18:22:59	1	6
TRIGGER 1 ACTIVATED			TRIGGER 1 ACTIVATED			TRIGGER 1 ACTIVATED		
18:18:38	17	7	18:22:24	21	7	18:22:59	13	6
TRIGGER 2 ACTIVATED			TRIGGER 2 ACTIVATED			TRIGGER 2 ACTIVATED		
18:18:39	12	123	18:22:24	7	167	18:23:00	14	78
TRIGGER 3 ACTIVATED			TRIGGER 3 ACTIVATED			TRIGGER 3 ACTIVATED		
18:18:39	10	123	18:22:24	8	167	18:23:00	11	78
18:18:39	10	123	18:22:24	10	167	18:23:00	9	78
18:18:39	10	123	18:22:25	9	167	18:23:00	9	78
18:18:39	9	123	18:22:25	10	167	18:23:00	9	78
18:18:39	9	123	18:22:25	9	167	18:23:01	9	78
18:18:39	10	123	18:22:25	9	167	18:23:01	9	78
18:18:40	9	123	18:22:25	9	167	18:23:01	9	78
18:18:40	9	123	18:22:25	9	167	18:23:01	9	78
18:18:40	9	123	18:22:25	9	167	18:23:01	9	78
18:18:40	9	7	18:22:26	9	6	18:23:01	9	6
FALL DETECTED!			FALL DETECTED!			FALL DETECTED!		

Figura 6.13 – Serie 1, esempio 2.

Dai dati presenti nella tabella in figura 6.13 si possono trarre diverse conclusioni che sono elencate di seguito:

- il valore  $acc$  utile alla soglia critica inferiore  $LFT_{acc}$  per l'attivazione del trigger1 è di 2g per la prima e la seconda caduta, 1g per la terza. Considerando che il valore imposto alla soglia critica inferiore è  $LFT_{acc} \leq 3g$  in questa fase di testing, non sono state apportate modifiche sul momento per permettere di analizzare meglio i falsi positivi, ma un'eventuale modifica della soglia potrebbe essere  $LFT_{acc} = 2g$ ;
- il valore  $acc$  utile alla soglia critica superiore  $UFT_{acc}$  per l'attivazione del trigger2 registra i valori di 17g per la prima caduta, 21g per la seconda caduta, 13g per la terza caduta. Considerando che il valore imposto in questa fase alla soglia critica superiore è  $UFT_{acc} \geq 11g$  e tenendo conto sia della natura simulata delle cadute (e quindi dell'accelerazione contenuta rispetto a una caduta vera) sia del fatto che durante i test la *sensibilità* raggiunge il valore massimo, si è deciso di impostare la soglia critica superiore a  $UFT_{acc} = 12$  per la successiva fase di test.
- il valore  $\omega$  utile alla soglia critica superiore  $UFT_{gyro}$  per l'attivazione del trigger3 registra i valori di 123°/s per la prima caduta, 167°/s per la seconda caduta, 78°/s per la terza caduta. Anche nella caduta 3, più blanda rispetto alle altre, il valore rientra ampiamente nel range definito per i test e non si verificano pertanto i presupposti per apportarvi modifiche.

- il valore  $\omega$  utile alla conferma dell'impatto rimane nel range definito dall'algoritmo che altro non è che il range di valori di equilibrio della velocità angolare. Variare i valori di questo range ha poco senso, in quanto sono i valori registrati dal giroscopio in condizioni di equilibrio.
- il momento di attesa, compreso tra l'attivazione del terzo trigger e l'effettivo rilevamento della caduta, evidenzia come il valore di  $\omega$  non venga ricalcolato mentre si aspetta l'impatto. Si è scelto di modificare questo aspetto nei test successivi in modo tale da monitorare  $\omega$  anche nella fase di attesa. Infatti, l'algoritmo di questa iterazione aspetta circa un secondo prima di controllare se sia avvenuto l'impatto, ma poiché ogni caduta ha una durata unica non è detto che essa si esaurisca dopo un secondo di attesa; pertanto, anche se generalmente le cadute non hanno durata maggiore al secondo, questo aspetto è stato approfondito nella seguente iterazione di test.

▪ Movimenti quotidiani (figura 6.14).

In esame due movimenti di "alzata/seduta" da una sedia, uno avente una velocità contenuta (alle ore 15:43:30) e uno caratterizzato da una maggiore velocità, per cercare di ingannare l'algoritmo, alle ore 15:43:39. Sono altresì presenti un terzo ed un quarto movimento in corrispondenza di momenti in cui il soggetto si è chinato a raccogliere un oggetto, rispettivamente alle ore 15:43:44 e 15:43:17.

L'attivazione di Trigger in corrispondenza del superamento di una delle soglie critiche, con relativo valore corrispondente, è evidenziata in verde, mentre la disattivazione degli stessi con relativi valori è evidenziata in rosso.

Sedia I			Sedia II			Chinarsi I			Chinarsi II		
Time(HH:MM:SS)	acc	$\omega$	Time(HH:MM:SS)	acc	$\omega$	Time(HH:MM:SS)	acc	$\omega$	Time(HH:MM)	acc	$\omega$
15:43:29	10	6	15:43:38	9	6	15:43:44	4	3	15:43:17	8	10
15:43:29	9	6	15:43:38	9	6				15:43:17	1	10
15:43:29	9	6	15:43:38	9	6	15:43:44	1	3			
15:43:29	12	6	15:43:38	10	6	TRIGGER 1 ACTIVATED			TRIGGER 1 ACTIVATED		
15:43:29	15	6	15:43:38	8	6						
15:43:30	12	6				15:43:44	13	3	15:43:17	6	10
15:43:30	8	6	15:43:39	2	6	TRIGGER 2 ACTIVATED			15:43:17	17	10
			TRIGGER 1 ACTIVATED						TRIGGER 2 ACTIVATED		
15:43:30	2	6				15:43:44	12	32			
TRIGGER 1 ACTIVATED			15:43:39	11	6	TRIGGER 3 ACTIVATED			15:43:17	18	22
			15:43:39	19	6				15:43:18	11	18
15:43:30	7	6	TRIGGER 2 ACTIVATED			15:43:44	12	40	15:43:18	13	20
15:43:30	11	6				15:43:44	11	54	15:43:18	13	14
15:43:30	8	6	15:43:39	15	11	15:43:45	11	60			
15:43:30	11	6	15:43:39	12	22	15:43:45	7	43	15:43:18	11	55
15:43:31	8	6	15:43:39	8	6	15:43:45	3	50	TRIGGER 3 ACTIVATED		
			15:43:40	9	6	15:43:45	7	54			
TRIGGER 1 DEACTIVATED						15:43:45	9	34	15:43:18	4	111
Non è stata rilevato un superamento della soglia critica superiore di acc nell'arco di 500ms			TRIGGER 2 DEACTIVATED			15:43:45	10	16	15:43:18	4	101
			TRIGGER 1 DEACTIVATED			15:43:45	9	9	15:43:18	9	56
			Non è stata rilevato un superamento della soglia critica superiore di $\omega$ nell'arco di 500ms						15:43:19	10	33
						15:43:45	9	5	15:43:19	10	18
						FALL DETECTED!			15:43:19	10	14
									15:43:19	9	9
									15:43:19	9	9
									15:43:19	9	5
									15:43:19	9	6
									FALL DETECTED!		

Figura 6.14 – Serie 1, esempio 3.

La tabella in figura 6.14 contiene gli ultimi dati raccolti per la prima serie di test e ci consente di trarre diverse conclusioni che verranno elencate di seguito:

- il movimento che corrispondente all'alzata e alla seduta dalla sedia è gestito correttamente dal dispositivo.

Il test "sedia I" consiste in una serie di movimenti naturali di alzata e seduta: in questi casi il valore della soglia critica inferiore  $LFT_{acc}$  viene superato ma quello della soglia critica superiore  $UFT_{acc}$  non viene soddisfatto.

Il test "sedia II" è invece relativo ad una serie di movimenti veloci di alzata e seduta: entrambe le soglie critiche dell'accelerazione vengono superate, ma la soglia critica  $UFT_{gyro}$  interviene a bloccare l'innescio di un falso positivo, poiché i movimenti presi in considerazione, indipendentemente dalla

velocità cui sono effettuati, non comportano una variazione significativa della velocità angolare del capo del soggetto, che mantiene quindi l'equilibrio;

- negli ultimi due test si verificano due falsi positivi in risposta al chinarsi del soggetto a velocità sostenuta per raccogliere un oggetto posizionato a terra, per poi rialzarsi. In questo caso i test si distinguono per la variazione della velocità cui sono state effettuate le azioni chinarsi-alzarsi, maggiore nel secondo caso.

Nel primo test ("Chinarsi I") il valore massimo della soglia critica superiore dell'accelerazione è  $UFT_{acc} = 13g$  ed i valori della velocità angolare sono  $32^\circ/s \leq \omega \leq 60^\circ/s$ . Visti anche i risultati dei precedenti esempi è stato quindi impostato il valore  $UFT_{acc} = 14g$  ed  $61^\circ/s \leq UFT_{gyro} \leq 400^\circ/s$  di modo da diminuire il valore di *sensibilità* dell'algoritmo, aumentando però la *precisione*.

- Il secondo test ("Chinarsi II") non è considerabile come fallimentare considerata la repentinità del movimento: vista la fascia di età cui si rivolge il dispositivo proposto, è accettabile presumere che movimenti simili non siano naturali e, in caso si verifichino, siano effettivamente ricollegabili ad una caduta del soggetto.

### Serie 2:

In questa fase si riassumono, senza entrare nel dettaglio, le tre categorie della precedente serie in una sola, prendendo in considerazione le misurazioni più rilevanti.

I valori delle soglie critiche durante la seconda serie di test sono elencati di seguito, evidenziando in **arancio** le variazioni registrate rispetto alla fase precedente:

- $LFT_{acc} = 2g$ , quindi  $trigger1 = true$  sse  $acc \leq 2g$
- $UFT_{acc} = 14g$ , quindi  $trigger2 = true$  sse  $acc \geq 14g$
- $61^\circ/s \leq UFT_{gyro} \leq 400^\circ/s$ , quindi  $trigger3 = true$  sse  $61^\circ/s \leq \omega \leq 400^\circ/s$

Utilizzando i valori sopra riportati è stato eseguito il secondo test (con una durata di venti minuti circa) durante il quale si è cercato di replicare i movimenti eseguiti durante la prima serie di test, avendo cura di annotare su un diario le azioni svolte e i valori corrispondenti delle variabili per il calcolo delle formule di *sensibilità* e *precisione*. Sono stati effettuati 25 test comprensivi di 10 cadute simulate e sono stati registrati i valori seguenti:

#TP (True positive) = 8  
#FN (False negative) = 2

#TN (True negative) = 11  
#FP (False positive) = 4

I valori sopra riportati hanno permesso il calcolo delle seguenti formule:

$$Sensitivity = \frac{\#TP}{\#TP + \#FN} = \frac{8}{8 + 2} = 0.8$$

$$Specificity = \frac{\#TN}{\#TN + \#FP} = \frac{11}{11 + 4} = 0.73$$

$$Accuracy = \frac{\#TN + \#TP}{\#TN + \#FN + \#TP + \#FP} = \frac{11 + 8}{11 + 2 + 8 + 4} = 0.76$$

Possiamo affermare che in questa seconda serie di test l'algoritmo abbia notevolmente migliorato la *specificità*, raggiungendo un valore di 0.73/1, registrando però un presumibile peggioramento nel valore della *sensibilità*, di 0.8/1. Il risultato è stato generalmente migliore rispetto a quello ottenuto nella precedente serie di test; infatti, la *precisione* ha raggiunto una percentuale del 76%.

Segue un'analisi completa e rapida dei dati registrati in *log.txt*, non soffermantesi sulle singole sezioni corrispondenti ai vari movimenti, ma analizzando la totalità dei movimenti, indicati nei vari titoli (*figure 6.15, 6.16*). L'attivazione di Trigger in risposta al superamento di una soglia critica, con relativo valore che l'ha causata, è evidenziata in **verde**, mentre la disattivazione degli stessi, con relativi valori, è evidenziata in **rosso**.

Chinarsi I			Chinarsi II			Sdraiarsi velocemente falso positivo		
Time(HH:MM:SS)	acc	$\omega$	Time(HH:MM:SS)	acc	$\omega$	Time(HH:MM:SS)	acc	$\omega$
11:13:03	7	5	11:13:39	11	5	11:14:13	23	19
11:13:04	13	5	11:13:39	11	5	11:14:13	4	19
11:13:04	16	5						
11:13:04	19	5	11:13:40	2	5	11:14:13	2	19
11:13:04	14	5	TRIGGER 1 ACTIVATED			TRIGGER 1 ACTIVATED		
11:13:04	19	5						
11:13:04	10	5	11:13:40	9	5	11:14:13	15	19
11:13:04	3	5	11:13:40	12	5	TRIGGER 2 ACTIVATED		
			11:13:40	10	5			
11:13:05	2	5				11:14:13	17	236
TRIGGER 1 ACTIVATED			11:13:40	14	5	TRIGGER 3 ACTIVATED		
			TRIGGER 2 ACTIVATED					
11:13:05	12	5				11:14:14	4	72
11:13:05	10	5	11:13:40	13	78	11:14:14	15	26
11:13:05	9	5	TRIGGER 3 ACTIVATED			11:14:14	9	44
11:13:05	9	5				11:14:14	9	14
11:13:05	9	5				11:14:14	10	13
11:13:05	9	5	11:13:41	16	38	11:14:14	9	10
11:13:06	10	5	11:13:41	18	207	11:14:14	9	10
			11:13:41	11	200	11:14:15	9	6
TRIGGER 1 DEACTIVATED			11:13:41	6	194	11:14:15	9	8
Non è stata rilevato un superamento della soglia critica superiore di acc nell'arco di 500ms			11:13:41	8	98			
						11:14:15	9	6
			TRIGGER 3 DEACTIVATED			FALL DETECTED!		
			TRIGGER 2 DEACTIVATED					
			TRIGGER 1 DEACTIVATED					
			Non è stata rilevato l'impatto in quanto $\omega$ non è tornato ai valori di equilibrio in 500ms					
			11:13:41	2	14			
			TRIGGER 1 ACTIVATED					
			11:13:41	9	29			
			11:13:42	8	18			
			11:13:42	13	25			
			11:13:42	9	19			
			TRIGGER 1 DEACTIVATED					
			Non è stata rilevato un superamento della soglia critica superiore di acc nell'arco di 500ms					

Figura 6.15 – Serie 2, esempio 1.

Caduta I			Caduta (più blanda) non rilevata		
Time(HH:MM:SS)	acc	$\omega$	Time(HH:MM:SS)	acc	$\omega$
11:12:36	9	7	11:15:26	9	6
			11:15:26	8	6
11:12:36	1	7			
TRIGGER 1 ACTIVATED			11:15:27	2	6
			TRIGGER 1 ACTIVATED		
11:12:37	13	7			
11:12:37	22	7	11:15:27	4	6
TRIGGER 2 ACTIVATED			11:15:27	14	6
			TRIGGER 2 ACTIVATED		
11:12:37	9	69	11:15:27	12	54
TRIGGER 3 ACTIVATED			11:15:27	13	25
			11:15:27	13	25
			11:15:27	10	28
11:12:37	4	76	11:15:28	8	34
11:12:37	3	160			
11:12:37	35	183	TRIGGER 2 DEACTIVATED		
11:12:37	9	98	TRIGGER 1 DEACTIVATED		
11:12:38	8	50	Non è stata rilevato un superamento della soglia critica superiore di $\omega$ nell'arco di 500ms		
11:12:38	11	42			
11:12:38	9	17			
11:12:38	9	8			
11:12:38	10	8			
11:12:42	9	5			
FALL DETECTED!					

Figura 6.16 – Serie 2, esempio 2.



Di seguito le conclusioni tratte dalla seconda serie di esempi: (figure 6.15, 6.16)

- in “*chinarsi I*” il soggetto si è chinato velocemente e si è poi arrestato: in questo caso il vincolo  $UFT_{acc} = 14g$  impostato a seguito della precedente serie di test permette di evitare un falso positivo, in quanto il valore massimo raggiunto dall’accelerazione in questo test è stato  $acc_{max} = 12$  dopo l’attivazione del primo trigger;
- in “*chinarsi II*” il soggetto si è chinato con velocità maggiore e si è poi rialzato repentinamente: nonostante tutte le soglie siano superate, la condizione di impatto finale non è soddisfatta perché a seguito delle azioni combinate il soggetto ha continuato a muoversi, perciò il dispositivo non registra alcuna caduta;
- in “*sdraiarsi velocemente*” si registra un falso positivo, probabilmente riconducibile all’elevata velocità con cui il soggetto ha eseguito l’azione. Distinguere un simile movimento da una caduta effettiva è però decisamente arduo visto che i dati prodotti dai moti considerati sono praticamente sovrapponibili;
- in “*caduta I*” si può apprezzare come le modifiche attuate a seguito della prima serie di test non abbiano inficiato il riconoscimento di una caduta standard. Di contro, in “*Caduta (più blanda)*”, questi nuovi valori non hanno permesso il riconoscimento dell’evento. Per ovviare tale inconveniente si è deciso di impostare il valore minimo di  $\omega$  a  $50^\circ/s$ .

A seguito dei test il codice dell’algoritmo è stato variato solo nei valori delle soglie critiche, di conseguenza non si ritiene necessario riportarlo nuovamente.

## 6.c – Conclusioni e prospettive di miglioramento

L’algoritmo responsabile del rilevamento della frequenza cardiaca ha raggiunto una buona capacità di stima della frequenza cardiaca. In un’ottica di miglioramento, l’utilizzo di un sensore qualitativamente superiore al Max30102, poco preciso, è una scelta sicuramente percorribile. Difatti, dai test eseguiti è facile vedere come il sensore Max30102 sia caratterizzato da un’elevata imprecisione, sicuramente dovuta al suo unico led, peraltro piuttosto debole, utilizzato per attuare la tecnica della fotoplethysmografia.

Relativamente all’algoritmo responsabile del rilevamento delle cadute, a seguito di ulteriori test si è deciso di impostare i seguenti valori alle soglie critiche (in **arancio** i valori variati rispetto alla precedente serie di test):

- $LFT_{acc} = 2g$ , quindi  $trigger1 = true \text{ sse } acc \leq 2g$
- $UFT_{acc} = 14g$ , quindi  $trigger2 = true \text{ sse } acc \geq 14g$
- $40^\circ/s \leq UFT_{gyro} \leq 400^\circ/s$ , quindi  $trigger3 = true \text{ sse } 40^\circ/s \leq \omega \leq 400^\circ/s$

Valori simili risultano in un leggero aumento della *sensibilità* (valore 0.83) e una scarsa diminuzione della *specificità* (valore 0.71) mantenendo stabile la *precisione*. Vista la fascia di età a cui gli occhiali sono indirizzati si è preferito propendere per la massima *sensibilità*, anche se a discapito della *specificità*, preferendo la segnalazione di un falso positivo piuttosto che l’omessa allerta a seguito di una caduta effettiva. In un’ottica di miglioramento, l’uso di un hardware più performante rispetto all’Arduino Nano 33 IoT e l’impiego di ulteriori sensori permetterebbero di ottenere risultati più soddisfacenti, permettendo ad ambo i valori di *precisione* e *sensibilità* di tendere ad 1.

Un’ulteriore prospettiva di miglioramento è infine l’utilizzo di un hardware generalmente più adatto e prestante: le limitazioni imposte dalla bassa qualità dei sensori e dalle limitate prestazioni dell’hardware sono state un grosso “bottleneck” nello sviluppo del prototipo.

## Capitolo 7.

# Conclusioni

L'elaborato presentato ha introdotto i "Sensor-equipped Eyeglasses", un paio di occhiali dotati di sensori ed indirizzati alla fascia più avanzata di età della popolazione, finalizzati ad offrire un tempestivo aiuto in caso di presentazione di una o più condizioni critiche che possano mettere a rischio il soggetto che ne fa utilizzo e in grado di permettere un monitoraggio costante dei suoi parametri vitali. A tal fine sono state affrontate alcune tra le più comuni problematiche collegate alla salute dell'essere umano, in modo da realizzare a livello hardware e software un dispositivo in grado di permettere la prevenzione e la segnalazione delle stesse. Si è progettato e sviluppato un primo prototipo hardware degli occhiali che, attraverso i sensori scelti ed alla scrittura di appositi algoritmi, si è rivelato in grado di svolgere alcuni tra i requisiti funzionali individuati, sfruttando il cloud fornito da Arduino. Gli algoritmi formulati sono stati quindi sottoposti ad una fase approfondita di test per migliorarne il funzionamento e l'affidabilità.

Tutti i codici scritti, compreso l'intero codice che governa gli "SE Eyeglasses", dai rilevamenti alla connessione al cloud Arduino, sono disponibili al seguente link: [GitHub](#).

### 7.a – Dal prototipo al prodotto finito

Come descritto più volte durante l'elaborato, il prototipo degli occhiali è stato creato senza alcuna conoscenza tecnica dei vari ambiti compresi nello sviluppo di un prodotto simile e avvalendosi di un hardware che, con le dovute eccezioni, non è assolutamente pensato per questo tipo di utilizzo (basti pensare che lo smartwatch Samsung Galaxy Watch 4, utilizzato per i confronti nel capitolo 6 dell'analisi dei dati della rilevazione della frequenza cardiaca, è dotato di un processore dual-core 1.2Ghz a 5nm e ben 1.5GB di RAM).

Con l'utilizzo di strumenti più appropriati gli "SE Eyeglasses" potrebbero affermarsi come un dispositivo paragonabile ai più blasonati smartwatch in commercio, presentando inoltre numerose funzionalità non disponibili nei classici wearable sul mercato. Nell'immagine seguente è rappresentata l'idea del progetto completo, comprensiva di hardware e funzionalità varie (figura 7.1).

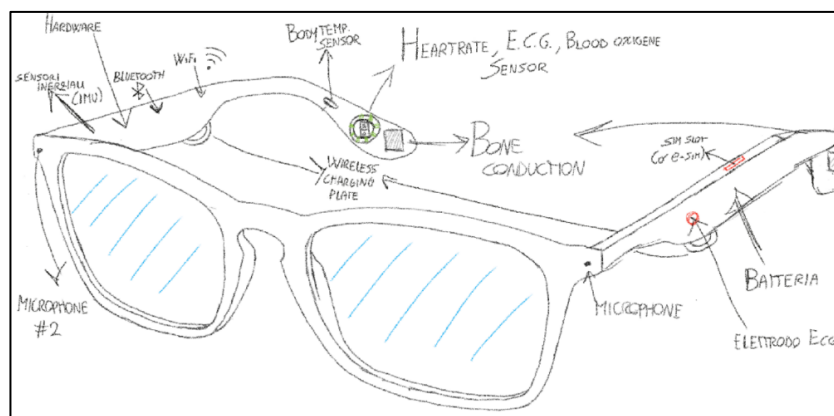


Figura 7.1 – SE Eyeglasses final product sketch prototype

### 7.b – Possibili applicazioni future

Un dispositivo come gli "SE eyeglasses" potrebbe avere un impatto non indifferente nella semplificazione e nella salvaguardia della vita degli anziani e delle categorie fragili in generale. È difficile anche solo immaginare gli svariati utilizzi che potrebbero derivare da un dispositivo del genere: l'impiego nelle RSA<sup>23</sup> è sicuramente un'ipotesi, ma gli "SE eyeglasses" sarebbero anche più utili se utilizzati da parte di anziani che vivono da soli, che potrebbero così essere monitorati da remoto da un tutore (o personale della RSA), in grado di controllarne i valori vitali e ricevere segnalazioni in caso di cadute accidentali. Ultima, ma non per importanza, la funzionalità audio integrata, che consentirebbe all'anziano, avvalendosi dell'assistente vocale, di controllare la propria frequenza cardiaca, effettuare chiamate ai propri cari, ascoltare musica, ecc.

<sup>23</sup> Residenze Sanitarie Assistenziali o più comunemente definite case di riposo e/o cura.



# Ringraziamenti

A conclusione di questo elaborato, ci terrei a ringraziare tutte quelle persone che, direttamente o indirettamente, hanno contribuito alla stesura della mia tesi ed al raggiungimento di questo importante traguardo della mia vita.

In primis desidero ringraziare la Prof.ssa Maria De Marsico, la mia relatrice, con la quale ho avuto il piacere di avere numerosi confronti durante lo svolgimento del tirocinio. La ringrazio per la sua disponibilità e la sua gentilezza, per la possibilità data di dare sfogo alla mia creatività ed alla mia fantasia, raggiungendo tutti gli obbiettivi preposti, merito del percorso universitario seguito ma soprattutto dei suoi preziosi consigli.

Ringrazio infinitamente la mia famiglia per avermi permesso di arrivare fin qui, per l'amore, la pazienza ed il sostegno nelle scelte difficili, fin da quella del mio percorso di studi.

Un ringraziamento speciale va alla mia fidanzata Giulia per essere stata sempre presente ed aver creduto in me sin dall'inizio. La ringrazio per avermi trasmesso la forza ed il coraggio per andare avanti e di essere la persona che più di tutte è stata capace di sostenermi nei momenti difficili. Dedico questo traguardo a noi.

Un grazie va a tutti i veri amici e a tutte quelle persone che mi hanno aiutato nel difficile percorso universitario e della vita in generale, fornendomi utili consigli e opinioni. La persona che sono è merito della vostra compagnia.

Ringrazio di cuore il mio amico e collega Simone che più di tutti mi ha aiutato nella stesura della tesi, dai test alle correzioni. La tua amicizia e aiuto è una garanzia di fratellanza da una vita.

Ringrazio dal profondo del cuore le mie care amiche Giada e Alessia che, con la loro immensa gentilezza e conoscenza, sono state di fondamentale aiuto nella stesura di questo testo.

Infine, mi congratulo con me stesso per la perseveranza e l'impegno, la tenacia ed i sacrifici, gli errori e le scelte sbagliate, tutte cose che mi hanno permesso di giungere fin qui.

Grazie a tutti, senza ognuno di voi non ce l'avrei fatta.

# Bibliografia

- [1] L. S. Vailshery, "Number of internet of things (IoT) connected devices worldwide in 2018, 2025 and 2030", 22 Jan. 2021, [Statista](#), [URL](#), [posizione citazione nel testo](#)
- [2] J. Steward, "The Ultimate List of Internet of Things Statistics for 2022", 5 Dec. 2021, [FindStack](#), [URL](#), [posizione citazione nel testo](#)
- [3] Eurostat, "Causes of death statistics – people over 65", Aug. 2021, [Eurostat Statistics Explained](#), [URL](#), [posizione citazione nel testo](#)
- [4] "Elettrocardiogramma", 4 Jan. 2022, [Wikipedia](#), [L'enciclopedia libera](#). Tratto il 13 Feb. 2022 da: [URL](#), [posizione citazione nel testo](#)
- [5] [World Health Organization](#), World Health Organization. Ageing, & Life Course Unit. (2008). "WHO global report on falls prevention in older age". World Health Organization, [URL](#), [posizione citazione nel testo](#)
- [6] Tinetti, M. E., Liu, W. L., & Claus, E. B. (1993). "Predictors and prognosis of inability to get up after falls among elderly persons". [Jama](#), 269(1), 65-70.; 06 Jan. 1993, [URL](#), [posizione citazione nel testo](#)
- [7] [Statista Research Department](#), "Share of adult consumers who currently wear eyeglasses in the U.S. in 2016, by age group", 05 Dec. 2016, [Statista](#), [URL](#), [posizione citazione nel testo](#)
- [8] Anderson, M., & Perrin, A. (2017). "Technology use among seniors. Washington, DC", [Pew Research Center for Internet & Technology](#), [URL](#), [posizione citazione nel testo](#)
- [9] [Google Immagini](#), [PNGWing](#), [PNGEgg](#), [Flaticon](#), [Wikipedia](#), [Pixabay](#), [Sociologiaonweb](#), [Pinterest](#) [Photo1](#); [Photo2](#); [Photo3](#); [Photo4](#); [Photo5](#); [Photo6](#); [Photo7](#); [Photo8](#); [posizione citazione nel testo](#)
- [10] [Arduino®](#), "Arduino Nano 33 IoT – Datasheet", 10 Feb. 2022, [URL](#), [posizione citazione nel testo](#)
- [11] Castaneda, D., Esparza, A., Ghamari, M., Soltanpur, C., & Nazeran, H. (2018). "A review on wearable photoplethysmography sensors and their potential future applications in health care". International journal of biosensors & bioelectronics, 4(4), 195. [PubMed Central®](#); 20 Mar. 2018, [URL](#), [posizione citazione nel testo](#)
- [12] Glodendance, "Bone conduction: How it Works" [Goldendance CO., Ltd](#), [URL](#), [posizione citazione nel testo](#)
- [13] SparkFun, "SparkFun MAX301x Particle Sensor Library", [GitHub](#), [URL](#), [posizione citazione nel testo](#)
- [14] Huynh, Q. T., Nguyen, U. D., Irazabal, L. B., Ghassemian, N., & Tran, B. Q. (2015). "Optimization of an accelerometer and gyroscope-based fall detection algorithm". Journal of Sensors, 2015. [Hindawi](#), [URL](#), [posizione citazione nel testo](#)
- [15] Repository GitHub del progetto di proprietà di Caciolo Valerio: [URL](#)