



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Meccanica

**Studio ed implementazione di un sistema
per misura di segnali PPG su scheda a
microcontrollore per il monitoraggio di
guidatori**

**Study and development of a
microcontroller-based acquisition system
of PPG signals for drivers' monitoring**

Tesi di laurea di:

Gaia Cingolani

Relatore:

Prof. Alessandro Terenzi

Correlatore

Dott. Ing. Valeria Bruschi

Prof.ssa Stefania Cecchi

Anno Accademico 2023-2024

*A chi ha perso la vita in un incidente stradale
e ai loro familiari.*

*Questa tesi è dedicata a voi,
che avete affrontato il dolore più grande
e l'assenza più profonda.*

*Che questa tesi possa essere un piccolo
ingranaggio nel grande motore del progresso,
un tributo a chi non è più con noi e un impegno
verso un futuro migliore,
dove la tecnologia e l'ingegneria lavorano
insieme,
per la vita.*

Indice

INDICE	3
INDICE DELLE FIGURE	5
ABSTRACT	7
INTRODUZIONE	8
CAPITOLO 1: VOLANTI SENSORIZZATI	10
1.1. FUNZIONI E SCOPI DEI VOLANTI INTELLIGENTI.....	10
1.2. SISTEMA DI VOLANTE SENSORIZZATO.....	11
1.3 UTILIZZO DEI SENSORI PPG NEL VOLANTE SENSORIZZATO.....	16
CAPITOLO 2: DISPOSITIVI E STRUMENTI UTILIZZATI	19
2.1 ARDUINO.....	19
2.1.1 PROGRAMMAZIONE CON ARDUINO: IDE	19
2.1.2 ANALISI DELL’HARDWARE DI ARDUINO UNO	20
2.1.3 APPLICAZIONI DI ARDUINO	22
2.2 SENSORI PPG: FUNZIONAMENTO GENERALE.....	22
2.3 SENSORE DI FREQUENZA CARDIACA CON GUSCIO: GROVE – FINGER-CLIP	24
2.3.1 COLLEGAMENTI DEL SENSORE GROVE – FINGER-CLIP ALLA SCHEDA ARDUINO UNO	26
2.3.1.1 PRIMO COLLEGAMENTO DEL SENSORE GROVE-FINGER-CLIP AL MICROCONTROLLORE: PIN ANALOGICI 26	
2.3.1.2 SECONDO COLLEGAMENTO DEL SENSORE GROVE-FINGER-CLIP AL MICROCONTROLLORE: PIN DIGITALI	27
2.4 PULSE SENSOR	28
2.4.1 FUNZIONAMENTO HARDWARE.....	28
2.4.2 COLLEGAMENTO DEL PULSE SENSOR ALL’ARDUINO UNO	29
2.5 SENSORE MAX30102	30
2.5.1 COLLEGAMENTO DEL SENSORE MAX30102 AD ARDUINO UNO.....	33
2.6 IL PULSOSSIMETRO BEURER.....	35
CAPITOLO 3: IMPLEMENTAZIONE E VALIDAZIONE DEI CODICI PER L’ACQUISIZIONE DELLA FREQUENZA CARDIACA	37
3.1 IMPLEMENTAZIONE DELL’ALGORITMO MATLAB PER L’ANALISI STATISTICA DELLE ACQUISIZIONI DI FREQUENZA CARDIACA	37
3.2 IMPLEMENTAZIONE DEI CODICI PER GROVE - FINGER-CLIP.....	39
3.2.1 PRIMO CODICE PER IL GROVE-FINGER-CLIP: COLLEGAMENTO TRAMITE I PIN ANALOGICI	40
3.2.2 SECONDO CODICE PER IL GROVE-FINGER-CLIP: COLLEGAMENTO TRAMITE I PIN DIGITALI	41
3.2.3 PROVE SPERIMENTALI PER VERIFICARE L’EFFICACIA DELL’ALGORITMO.....	42
3.3 IMPLEMENTAZIONE DEI CODICI PER PULSE SENSOR	45
3.3.1 PRIMO CODICE: TRACCIAMENTO DEL BATTITO CARDIACO	46
3.3.1.1 PROVE SPERIMENTALI PER VERIFICARE L’EFFICACIA DEL PRIMO CODICE PER IL PULSE SENSOR	48

3.3.2	SECONDO CODICE: MISURAZIONE DELLA FREQUENZA CARDIACA.....	49
3.3.2.1	PROVE SPERIMENTALI PER VERIFICARE L'EFFICACIA DEL SECONDO CODICE PER IL PULSE SENSOR	49
3.3.3	TERZO CODICE: SOFTWARE "PROCESSING VISUALIZER"	50
3.3.3.1	PROVE SPERIMENTALI PER VERIFICARE L'EFFICACIA DEL PROCESSING VISUALIZER	51
3.4	SENSORE MAX30102: UTILIZZO DELLA LIBRERIA "SPARK FUN ELECTRONICS" E COLLEGAMENTI DEL SENSORE.....	53
CAPITOLO 4: PROVE E RISULTATI SPERIMENTALI SUL VOLANTE INTELLIGENTE		56
4.1	INSTALLAZIONE DEL SENSORE GROVE – FINGER-CLIP, CON INVOLUCRO, SUL VOLANTE.....	56
4.1.1	RISULTATI SPERIMENTALI OTTENUTI DAL GROVE – FINGER-CLIP CON INVOLUCRO	57
4.2	INSTALLAZIONE DEL SENSORE GROVE-FINGER-CLIP, SENZA INVOLUCRO, SU UN VOLANTE PROVVISORIO.....	59
4.2.1	RISULTATI SPERIMENTALI OTTENUTI DAL GROVE-FINGER-CLIP SENZA INVOLUCRO.....	60
CAPITOLO 5: DISCUSSIONE DEI RISULTATI, CONCLUSIONI E SVILUPPI FUTURI.....		64
BIBLIOGRAFIA		67

Indice delle Figure

Figura 1: Configurazione del sistema del volante sensorizzato.	12
Figura 2: Posizionamento dei sensori.	13
Figura 3: Volante sensorizzato realizzato.	15
Figura 4: Dati di rilevamento misurati dai sensori.	15
Figura 5: Il segnale PPG misurato dal BIOPAC MP36 (in alto) e il segnale rilevato dal sensore sul volante (al centro). La comparazione tra i due segnali PPG misurati dal BIOPAC (blu) e dal volante intelligente (rosso) è mostrata nel grafico in basso.	16
Figura 6: Dati acquisiti dai sensori del volante sensorizzato.	17
Figura 7: Scheda Arduino UNO.	20
Figura 8: Posizionamento del dito sui sensori ottici di frequenza cardiaca.	22
Figura 9: Andamento del segnale nel tempo.	23
Figura 10: Sensore Grove – Finger-clip.	24
Figura 11: Piedinatura del sensore Grove-Finger-Clip.	26
Figura 12: Primo collegamento del sensore Grove Finger-Clip alla scheda.	26
Figura 13: Secondo collegamento del sensore Grove-Finger-Clip alla scheda.	27
Figura 14: Pulse sensor.	28
Figura 15: Componenti del Pulse Sensor.	28
Figura 16: Specifiche tecniche del Pulse sensor.	29
Figura 17: Piedinatura e cablaggio del Pulse Sensor.	29
Figura 18: Sensore di frequenza cardiaca MAX30102.	30
Figura 19: Panoramica dell'hardware del sensore MAX30102.	30
Figura 20: Spettro di assorbimento dell'emoglobina ossigenata e deossigenata.	31
Figura 21: Vista anteriore e posteriore della scheda a circuito stampato (PCB) del sensore MAX30102.	32
Figura 22: Specifiche Tecniche del sensore MAX30102.	33
Figura 23: Piedinatura del sensore MAX30102.	33
Figura 24: Cablaggio tra il sensore MAX30102 e la scheda Arduino UNO.	34
Figura 25: Pulsossimetro Beurer.	35
Figura 26: Sensore e pulsossimetro indossati.	35
Figura 27: Codice MATLAB per creare il grafico Tempo-Frequenza cardiaca [s-bpm].	38
Figura 28: Primo codice per il sensore Grove-Finger-Clip.	40
Figura 29: Secondo codice per il sensore Grove-Finger-Clip.	41
Figura 30: Acquisizioni con il sensore Grove-Finger-Clip su donna di 22 anni dopo la corsa.	43
Figura 31: Acquisizioni con il sensore Grove-Finger-Clip su donna di 47 anni dopo la corsa.	44
Figura 32: Acquisizioni con il sensore Grove-Finger-Clip su un uomo di 25 anni a riposo.	44
Figura 33: Acquisizione su soggetto in condizioni non ottimali con il sensore Grove-Finger-Clip.	45
Figura 34: Primo codice per il Pulse Sensor.	46
Figura 35: Tracciato del battito cardiaco nel tempo rilevato dal lobo dell'orecchio tramite il Pulse Sensor.	48
Figura 36: Tracciato del battito cardiaco nel tempo rilevato dal polso carotideo tramite il Pulse Sensor.	48
Figura 37: Secondo Codice per il Pulse Sensor.	49
Figura 38: Codice per l'uso di Processing Visualizer.	50
Figura 39: Tracciato del battito cardiaco rilevato dal pollice tramite il Pulse Sensor.	51
Figura 40: Tracciato del battito cardiaco rilevato dall'indice tramite il Pulse Sensor.	52
Figura 41: Tracciato del battito cardiaco rilevato dal lobo dell'orecchio tramite il Pulse Sensor.	52
Figura 42: Tracciato del battito cardiaco rilevato dal polso carotideo tramite il Pulse Sensor.	52
Figura 43: Primo tentativo di collegamento per il MAX30102.	53
Figura 44: Secondo tentativo di collegamento per il MAX30102.	54
Figura 45: Terzo tentativo di collegamento per il MAX30102.	55
Figura 46: Installazione del sensore Grove-Finger-Clip sul volante.	56
Figura 47: Simulazione di guida con la periferica di gioco.	57
Figura 48: Acquisizione tramite Grove-Finger-Clip su Soggetto 1 alla guida.	57
Figura 49: Acquisizione tramite Grove-Finger-Clip su Soggetto 2 alla guida.	58

Figura 50: Sensore Grove-Finger-Clip senza involucro.	59
Figura 51: Posizionamento del Grove-Finger-Clip, senza guscio, per poggiare il pollice.....	60
Figura 52: Posizionamento del Grove-Finger-Clip, senza guscio, per poggiare il palmo della mano.	60
Figura 53: Monitoraggio del battito cardiaco dal pollice tramite il sensore Grove-Finger-Clip privo di involucro.....	61
Figura 54: Test con Sensore PPG a contatto con il pollice.	61
Figura 55: Monitoraggio del battito cardiaco dal palmo della mano tramite il sensore Grove-Finger-Clip privo di involucro.	62
Figura 56: Test con Sensore PPG a contatto con il palmo della mano.	62
Figura 57: Andamento temporale della frequenza cardiaca durante l'interazione con il videogioco "F1 23" del Soggetto 1.	63
Figura 58: Andamento temporale della frequenza cardiaca durante l'interazione con il videogioco "F1 23" del Soggetto 2.	63

Abstract

Negli ultimi anni, la crescente attenzione verso la sicurezza stradale e la salute dei conducenti ha stimolato lo sviluppo di volanti sensorizzati. Il monitoraggio continuo delle condizioni fisiologiche dei guidatori è cruciale per prevenire incidenti stradali causati da sonnolenza o distrazione. Questa tesi presenta lo sviluppo e l'implementazione di un sistema basato su un microcontrollore per la misura dei segnali fotoplethysmografici (PPG), utilizzati per monitorare la frequenza cardiaca. Il sistema proposto integra i sensori PPG con una scheda a microcontrollore, permettendo l'acquisizione e l'elaborazione in tempo reale dei dati fisiologici del guidatore.

Sono stati esaminati tre sensori ottici di frequenza cardiaca: il Grove-Finger-Clip, il Pulse Sensor e il sensore MAX30102. I test effettuati hanno valutato la precisione e la reattività di ciascun sensore in condizioni operative simulate. I risultati ottenuti hanno permesso di identificare il sensore più performante, tenendo conto delle specifiche esigenze di integrazione e delle condizioni di utilizzo previste per il volante sensorizzato.

Il presente lavoro getta le basi per lo sviluppo di un sistema di monitoraggio della frequenza cardiaca integrato in un volante intelligente, evidenziando sia le potenzialità che le limitazioni dei sensori testati. Le future ricerche dovranno concentrarsi sull'ottimizzazione del sensore scelto e sull'esplorazione di nuove tecnologie per migliorare ulteriormente l'esperienza dell'utente, non solo in ambito ludico, ma anche in contesti clinici e di monitoraggio della salute.

Introduzione

Il presente lavoro di tesi si concentra su lo studio, l'implementazione e la verifica sperimentale di sensori di frequenza cardiaca integrati su una scheda a microcontrollore. In particolare, sono stati esaminati sensori fotoplethysmografici (PPG) destinati all'installazione su volanti intelligenti, con l'obiettivo di monitorare i parametri vitali degli individui alla guida di veicoli.

La sicurezza stradale rappresenta una delle principali preoccupazioni a livello globale, e la sonnolenza e lo stress del conducente sono fattori significativi negli incidenti stradali. L'integrazione di sensori di frequenza cardiaca nei volanti può fornire un monitoraggio continuo e non invasivo dei parametri vitali del conducente, contribuendo a prevenire situazioni di pericolo e migliorando la sicurezza complessiva. I sensori PPG, utilizzati per rilevare la frequenza cardiaca, forniscono dati preziosi per valutare lo stato di sonnolenza o stress dell'individuo. In genere, la pulsazione cardiaca diminuisce quando una persona è assennata o sta per addormentarsi. Al contrario, quando un individuo è stressato o agitato, il rilascio di ormoni come l'adrenalina provoca un aumento del battito cardiaco. Lo stress può anche aumentare la pressione sanguigna, influenzando indirettamente la frequenza cardiaca rilevata dai sensori PPG.

Nel corso di questa tesi, sono stati analizzati tre sensori ottici di frequenza cardiaca: il Grove-Finger-Clip, il Pulse Sensor e il sensore MAX30102. I tre sensori sono stati collegati alla scheda Arduino Uno, e per ciascun sensore sono stati verificati gli opportuni sketch sul software di Arduino. Successivamente, sono stati effettuati i test per valutare la precisione e la reattività di ciascun sensore in condizioni operative simulate. I risultati ottenuti hanno permesso di identificare il sensore più performante, tenendo conto delle specifiche esigenze di integrazione e delle condizioni di utilizzo previste per il volante sensorizzato.

La presente tesi è articolata in cinque capitoli principali.

Nel primo capitolo, viene fornita una panoramica dei modelli di volanti intelligenti, con un'analisi dettagliata delle diverse modalità di realizzazione del sistema. Questo capitolo introduce i concetti fondamentali e le tecnologie alla base dei volanti intelligenti.

Il secondo capitolo si concentra sul sistema hardware, descrivendo i dispositivi e i sensori utilizzati per lo sviluppo del presente progetto di tesi. Viene illustrata l'architettura del sistema e le specifiche tecniche dei componenti impiegati.

Nel terzo capitolo, vengono presentati i vari algoritmi sviluppati durante lo studio. Questo capitolo include una descrizione delle prove sperimentali condotte per verificare l'efficacia degli algoritmi e i risultati ottenuti.

Il quarto capitolo descrive l'installazione del sensore Grove-Finger-Clip sul volante, sia con che senza involucro protettivo. Vengono inoltre riportate le acquisizioni dei dati ottenute durante i test, con un'analisi dei risultati.

Infine, nel quinto capitolo, si tirano le somme del lavoro svolto. Vengono discussi i risultati ottenuti, evidenziando i punti di forza e le aree di miglioramento, e si delineano i possibili sviluppi futuri del progetto.

Capitolo 1: Volanti Sensorizzati

In questo capitolo viene fornita una panoramica dei modelli di volanti intelligenti, con un'analisi dettagliata delle diverse modalità di realizzazione del sistema. Questo capitolo introduce i concetti fondamentali e le tecnologie alla base dei volanti intelligenti.

1.1. Funzioni e scopi dei volanti intelligenti

Negli ultimi anni, la crescente attenzione verso la sicurezza stradale e la salute degli individui ha stimolato lo sviluppo di sistemi di monitoraggio all'interno delle automobili. La possibilità di integrare sensori avanzati nei veicoli ha spinto ricercatori e ingegneri a esplorare nuove soluzioni, come lo sviluppo di volanti intelligenti. Questi volanti mirano a migliorare la sicurezza stradale e a fornire un monitoraggio continuo e non invasivo della salute del conducente, affrontando due problemi principali.

Da un lato, l'aumento degli incidenti stradali causati dalla stanchezza e dallo stress del guidatore. La "US National Sleep Foundation" ha riportato che 1,9 milioni di guidatori hanno avuto un incidente o un quasi incidente a causa della sonnolenza. Inoltre, il 51% dei conducenti intervistati ha ammesso di aver guidato mentre era assennato e il 17% ha riferito di essersi effettivamente addormentato al volante [1].

Dall'altro lato, questi volanti sensorizzati affrontano la necessità di sviluppare la medicina personalizzata, rilevando e trattando precocemente le malattie croniche, in largo aumento a causa dell'uso di tabacco, dell'inattività fisica, del consumo dannoso di alcol e delle diete non salutari. Le malattie croniche uccidono 41 milioni di persone ogni anno, rappresentando il 71% di tutti i decessi a livello mondiale. Il loro rilevamento precoce è cruciale, motivo per cui si è pensato di sviluppare una piattaforma di monitoraggio nelle auto, uno dei luoghi in cui passiamo più tempo [2].

Diversi studi hanno dimostrato che gli elettrodi per l'acquisizione di elettrocardiogramma (ECG) possono essere integrati in vari punti dell'auto, come il volante, il sedile o lo schienale. Tuttavia, il volante ha un vantaggio decisivo perché i segnali più importanti possono essere monitorati dalle mani del conducente che sono a contatto con il volante. Per questo motivo, è preferibile inserire i sensori sul volante piuttosto che in altre parti del veicolo.

1.2. Sistema di volante sensorizzato

Per questo progetto di tesi sono stati analizzati due articoli scientifici che trattano l'implementazione e la progettazione di sistemi di volanti intelligenti. Sebbene entrambi gli articoli affrontino tematiche simili, presentano approcci e applicazioni leggermente differenti.

Il primo articolo, intitolato **“Smart Steering Wheel System for Driver’s Emergency Situation using Physiological Sensors and Smart Phone”** [1], si focalizza principalmente sulla gestione delle emergenze e sulla prevenzione della sonnolenza del conducente in modo non invasivo. Il secondo articolo, **“Design of Smart Steering Wheel for Unobtrusive Health and Drowsiness Monitoring”** [2], pubblicato sulla rivista “Sensors”, mira a un monitoraggio più ampio della salute del conducente, includendo anche il possibile rilevamento di malattie cardiovascolari.

Il primo articolo approfondisce lo sviluppo di un sistema di allarme e gestione delle emergenze tramite comunicazione via Bluetooth e CAN, che verrà descritto più avanti nel capitolo.

Al contrario, il secondo articolo descrive più in dettaglio i sensori fisiologici ECG e PPG, implementati anche nel modello di volante del primo articolo. Inoltre, il volante descritto nella rivista “Sensors” è dotato di un’unità di misura inerziale (IMU), composta da un accelerometro e un giroscopio, per rilevare i movimenti del volante e identificare la stanchezza del conducente. I dati del giroscopio e dell’accelerometro vengono utilizzati per rilevare artefatti e movimenti nei segnali PPG e ECG dovuti alla rotazione del volante e alle irregolarità della strada, come buche e dossi. Questo articolo descrive anche le prove pratiche, eseguite sia in laboratorio che in condizioni reali. Nei test in laboratorio il volante è stato fissato su una console con inclinazione regolabile per simulare le condizioni di guida reali. Sono stati misurati i segnali ECG, PPG e IMU e confrontati con dispositivi commerciali per verificarne la precisione. Nei test in condizioni reali il volante è stato montato su un’auto (Volkswagen Sharan) e testato su strade senza traffico per garantire la sicurezza del conducente. I dati raccolti sono stati analizzati per valutare la funzionalità del sistema in situazioni di guida quotidiana.

Ora verrà presentato, in dettaglio, il modello di volante sensorizzato sviluppato nel primo articolo [1], per fornire una visione più concreta delle caratteristiche tecniche di questi sistemi.

La configurazione del sistema realizzato è rappresentata nella Figura 1 [1].



Figura 1: Configurazione del sistema del volante sensorizzato.

Il sistema di volante intelligente finanziato dal **R&D Program of MKE** (Ministry of Knowledge Economy) è composto da tre parti principali:

A. **I sensori fisiologici**, che sono tre sensori che rilevano lo stato del conducente: ECG, PPG (pulsossimetro) e sensore di pressione.

I sensori ECG e PPG misurano la frequenza cardiaca del conducente. Come mostrato nella Figura 2, i sensori ECG sono posizionati sul lato sinistro e destro del volante, mentre i sensori PPG, anch'essi situati a destra e sinistra, sono composti da tre parti principali: la parte del sensore, del filtro e dell'amplificatore. Infine il sensore di pressione, montato sulla parte inferiore del volante, rileva la forza di presa del conducente.

B. **L'ECU (Unità di controllo elettronico)** è collegata ai sensori fisiologici, al sistema di allarme e allo smartphone. Ha due funzioni principali:

- 1) Comunicare con il cellulare tramite Bluetooth per trasmettere i dati sullo stato del conducente, oltre a ricevere e trasmettere dati sullo stato del veicolo tramite la rete CAN.
 - 2) Controllare il sistema di allarme, che include avvisi tramite vibrazione, luci e segnali acustici, in base al livello di sonnolenza del conducente.
- C. **L'app** che consente di consultare e trasmettere le informazioni sulle condizioni del conducente. In caso di necessità invia SMS al centro di emergenza e gestisce il sistema di allarme per avvisare il guidatore.

Le coordinate della vettura vengono individuate utilizzando il GPS dello smartphone e, in caso di segnale debole, la rete CDMA.

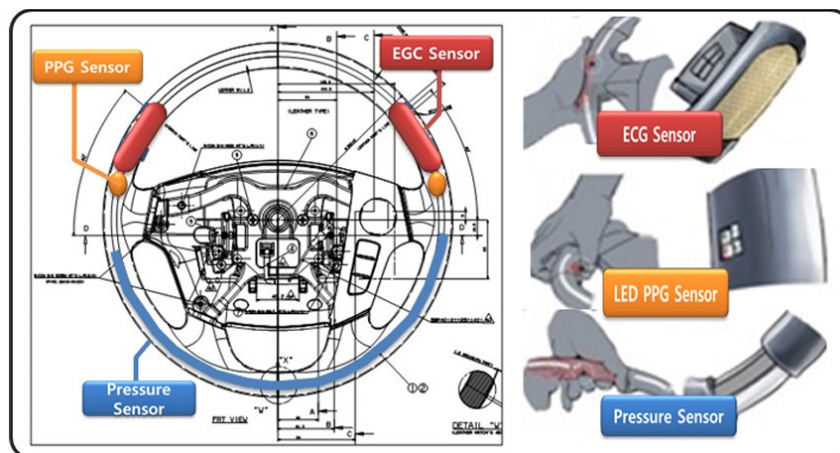


Figura 2: Posizionamento dei sensori.

Implementazione del Sistema di Volante Intelligente

Questa sezione descrive come il sistema di volante intelligente sia progettato per migliorare la sicurezza del conducente monitorando il suo stato fisiologico e fornendo avvisi tempestivi in caso di sonnolenza.

A. Progettazione del Volante Intelligente

1. **Posizionamento dei Sensori:** Il volante descritto nell'articolo [1] è progettato per essere impugnato alle ore 9 e 3, in modo da evitare che le braccia del conducente vengano colpite dall'airbag interno al volante in caso di apertura. I sensori ECG sono posizionati all'esterno del volante, i PPG all'interno, e i sensori di pressione nella parte inferiore.

2. **Cablaggio con Molla a Spirale:** Per evitare che i fili elettrici si attorciglino quando il volante viene girato, viene utilizzata una molla a spirale situata sotto il modulo del cruscotto. Questa molla mantiene la connessione elettrica sicura tra i sensori fisiologici e l'ECU.

B. Sistema di Allarme

Quando il conducente è in uno stato di sonnolenza di basso livello, l'ECU controlla il sistema di avviso per allertarlo. Il sistema di allarme è composto da tre segnali principali:

1. **Sistema di Allarme con LED:** Una banda LED flessibile è montata davanti al sedile del conducente e lampeggia con una frequenza di 1 Hz quando il livello di sonnolenza è basso.
2. **Sistema di Allarme con Vibrazione:** Due motori a vibrazione sono montati sul volante e attivati dall'ECU quando i sensori fisiologici rilevano uno stato di sonnolenza del conducente.
3. **Sistema di Allarme Sonoro:** Un cicalino meccanico è installato accanto all'altoparlante della portiera del conducente e attivato dall'ECU.

C. Sistema di Comunicazione del volante intelligente

Viene descritto in questa sotto sezione come il sistema di volante intelligente utilizza le reti CAN e Bluetooth per monitorare e comunicare i dati fisiologici del conducente, migliorando la sicurezza e l'efficienza del sistema complessivo.

1. **Sistema di Comunicazione CAN:** La rete CAN è utilizzata per la comunicazione tra l'ECU (Unità di Controllo Elettronico) del volante intelligente e il resto del veicolo. Questo sistema permette di trasmettere al veicolo dati critici come la velocità, la posizione del cambio, i dati sullo stato dei sensori fisiologici e del sistema di allarme.
2. **Sistema di Comunicazione Bluetooth:** La comunicazione tra l'ECU e lo smartphone avviene tramite Bluetooth, con un processo di accoppiamento automatico per mantenere la connessione e aggiornare i dati ogni 50 ms.

Implementazione e risultati sperimentali

La Figura 3 mostra il volante intelligente realizzato dal team supportato dal **R&D Program of the MKE** alla fine del lavoro di ricerca [1]. Come descritto in precedenza i due sensori ECG sono montati sui lati sinistro e destro del volante, mentre il sensore ottico PPG è stato montato sul lato sinistro del volante. Il sensore di presa è integrato sulle parti inferiori.

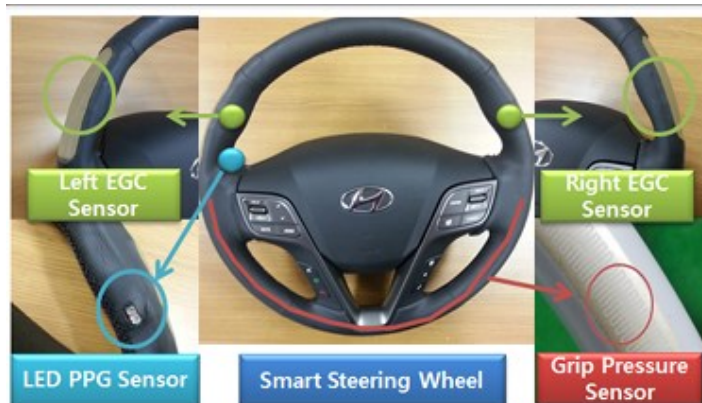


Figura 3: Volante sensorizzato realizzato.

I dati di rilevamento misurati dai sensori fisiologici del volante intelligente sono mostrati nella Figura 4.

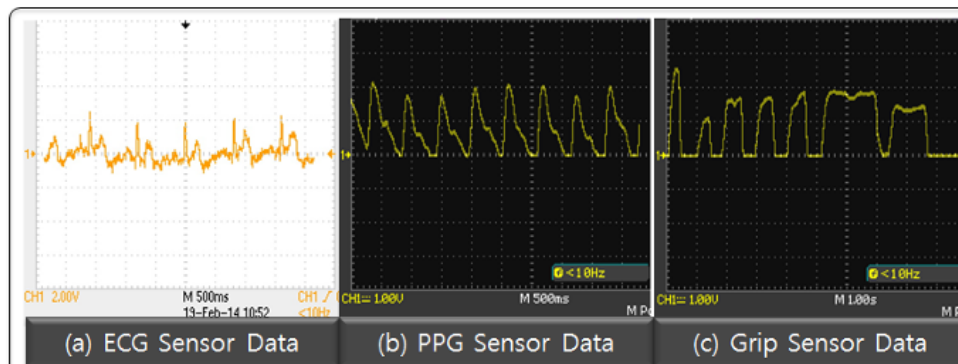


Figura 4: Dati di rilevamento misurati dai sensori.

1.3 Utilizzo dei Sensori PPG nel Volante Sensorizzato

In questo paragrafo, viene approfondito l'uso dei sensori PPG nel contesto del volante sensorizzato, come descritto nell'articolo pubblicato della rivista "Sensors" [2].

L'attenzione è rivolta alle deduzioni e conclusioni tratte dall'articolo a seguito delle prove sperimentali eseguite con i sensori PPG installati sul volante.

Nell'articolo della rivista "Sensors" il sensore PPG utilizzato per l'implementazione del volante sensorizzato è il MAX30102. Questo sensore, utilizzato e descritto anche nella presente tesi, è un pulsossimetro ad alta sensibilità per il monitoraggio della salute tramite dispositivi indossabili. Le caratteristiche tecniche e le funzionalità del MAX30102 sono dettagliate nel paragrafo 2.5.

Nell'articolo della rivista "Sensors" viene descritto il metodo utilizzato per verificare la corretta misurazione del segnale PPG da parte del sensore installato sul volante intelligente. Per questa verifica, l'utente indossava il dispositivo BIOPAC MP36 fissato al pollice destro tramite una fascia elastica, mentre il pollice sinistro era a contatto liberamente con il sensore MAX30102 montato sul volante.

In Figura 5 sono mostrati i segnali PPG rilevati dal BIOPAC MP36 e dal sensore installato sul volante intelligente.

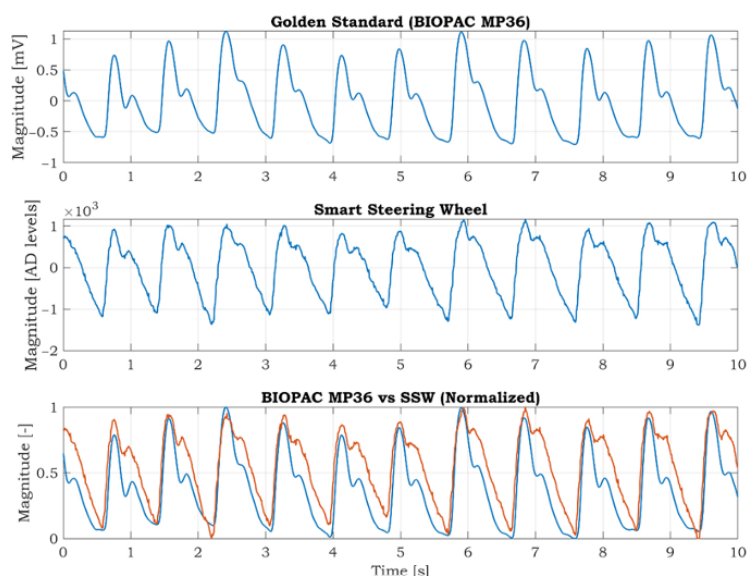


Figura 5: Il segnale PPG misurato dal BIOPAC MP36 (in alto) e il segnale rilevato dal sensore sul volante (al centro). La comparazione tra i due segnali PPG misurati dal BIOPAC (blu) e dal volante intelligente (rosso) è mostrata nel grafico in basso.

Dai grafici è evidente come il dito non fissato produce più artefatti nel segnale, quindi la qualità è inferiore rispetto alle misurazioni del BIOPAC.

Inoltre, il segnale PPG misurato sulle dita di mani diverse può variare perché il sistema dei vasi sanguigni non è simmetrico e dipende dalle condizioni di salute del soggetto. Nonostante questi fatti, si può concludere che le posizioni dei picchi sono quasi le stesse, come si vede nel grafico in basso della Figura 5. Le posizioni dei picchi PPG possono essere utilizzate per calcolare i battiti al minuto (BPM) o la velocità dell'onda del polso se il segnale ECG è noto.

Nell'articolo della rivista "Sensors" viene poi fatta un'analisi preliminare dei dati per dimostrare l'usabilità delle acquisizioni eseguite dai sensori installati sul volante intelligente. È stato notato che il segnale PPG nel canale a infrarossi (IR) ha una qualità migliore rispetto al canale rosso (RD) (Figura 6) perché la luce rossa penetra a una profondità minore rispetto agli infrarossi e fornisce meno informazioni sul sangue arterioso.

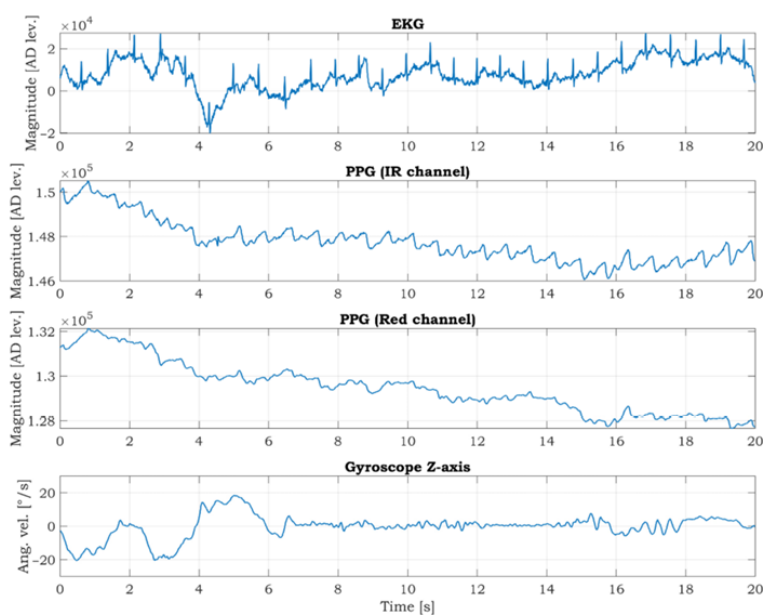


Figura 6: Dati acquisiti dai sensori del volante sensorizzato.

Si presume che una parte degli artefatti di movimento nei segnali sia causata dalla rotazione del volante sensorizzato. Per questo motivo, nell'articolo, è stato selezionato

l'intervallo di tempo da 12 e 14 secondi, durante il quale il volante intelligente non stava ruotando, come indicato dai dati giroscopici.

Gli artefatti di movimento (MA) rappresentano un problema significativo nella misurazione dei segnali PPG e nella successiva stima della saturazione di ossigeno nel sangue (SpO₂). Questi artefatti sono causati da qualsiasi movimento relativo tra la pelle e il sensore PPG, accompagnato da variazioni nell'accoppiamento ottico tra il tessuto illuminato e il sensore.

La soluzione generalmente utilizzata per ridurre gli artefatti di movimento consiste nel fare la media dei dati rumorosi su un periodo di tempo più lungo e mantenere i dati fino a quando non vengono qualificati come puliti.

Nel caso dei volanti intelligenti, il problema degli artefatti di movimento nel segnale PPG è particolarmente critico perché il sensore ottico non è fissato al dito. Per aumentare l'affidabilità della misurazione del battito cardiaco e della saturazione del sangue, è altamente consigliabile rimuovere i segmenti di segnale rumorosi acquisiti durante i rapidi movimenti del volante. I dati giroscopici possono identificare facilmente questi segmenti rumorosi, rendendo l'installazione del giroscopio sul volante sensorizzato estremamente importante [2].

Capitolo 2: Dispositivi e Strumenti Utilizzati

In questo capitolo vengono descritti il sistema hardware, i dispositivi e i sensori utilizzati per lo sviluppo del presente progetto di tesi.

2.1 Arduino

Arduino è una piattaforma a licenza libera di prototipazione elettronica basata su hardware e software facili da usare. È un progetto italiano che ha preso avvio nel 2005 e ha portato una rivoluzione nel campo dell'elettronica proprio per la sua natura open-source. Ciò significa che sia il software che l'hardware sono accessibili e modificabili da chiunque. Gli utenti possono scaricare gli schemi delle schede, apportare modifiche e creare le proprie versioni senza dover pagare licenze. Questo approccio ha stimolato una vasta comunità di sviluppatori e hobbisti che condividono progetti e miglioramenti. La piattaforma Arduino è composta da una scheda a microcontrollore e da un ambiente di sviluppo integrato (IDE) che utilizza un linguaggio di programmazione semplificato basato su Wiring [3]. Arduino comprende sia una componente software che una componente hardware, che verranno descritte nei paragrafi successivi.

2.1.1 Programmazione con Arduino: IDE

La programmazione di Arduino avviene tramite l'IDE di Arduino, che permette di scrivere, compilare e caricare il codice sulla scheda. Il linguaggio di programmazione è basato su C/C++ e include funzioni specifiche per interagire con l'hardware. Un programma Arduino, chiamato "sketch", è composto da due funzioni principali:

- **setup ():** Funzione eseguita una sola volta all'avvio del programma, viene utilizzata per inizializzare variabili, pin e librerie, che rimarranno invariati durante l'esecuzione.
- **loop ():** Funzione eseguita ripetutamente, contiene il codice principale del programma.

2.1.2 Analisi dell'Hardware di Arduino Uno

Le schede Arduino sono disponibili in vari modelli, ognuno con caratteristiche specifiche. I modelli più comuni includono:

- **Arduino Uno:** La scheda più popolare, ideale per principianti. È basata sul microcontrollore ATmega328P.
- **Arduino Mega:** Offre più pin di input/output e memoria rispetto all'Uno, ed è basata sul microcontrollore ATmega2560.
- **Arduino Nano:** Una versione compatta dell'Uno, ideale per progetti con ingombro limitato.

Per questo progetto di tesi, è stata utilizzata la scheda **Arduino Uno**, che verrà ora analizzata nel dettaglio.

Arduino Uno è una scheda che dispone di 14 pin digitali I/O, programmabili come ingressi o uscite, e di 6 ingressi per l'acquisizione e l'elaborazione di segnali analogici. La scheda è dotata di una porta USB femmina per la comunicazione e di un connettore femmina per l'ingresso dell'alimentazione. La scheda si presenta come mostrato in Figura 7 [4]:

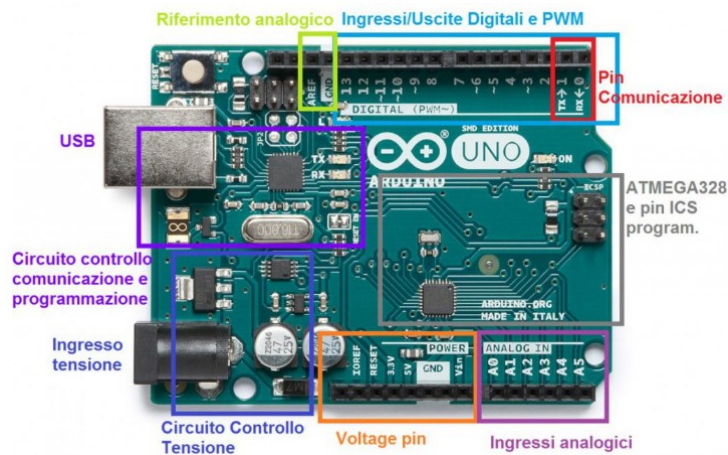


Figura 7: Scheda Arduino UNO.

Pin di alimentazione (Voltage Pin o Power):

- **IOREF:** Questo pin fornisce la tensione di alimentazione del microcontrollore. Se il microcontrollore è alimentato a 3.3 V, anche su questo pin la tensione sarà

di 3.3 V, permettendo a tutti i circuiti di operare alla stessa tensione del microcontrollore.

- **RESET:** Collegato al pin di reset del microcontrollore, consente di resettare il microcontrollore dall'esterno o di resettare altri componenti quando il microcontrollore viene resettato.
- **3.3 V e 5 V:** Questi pin forniscono rispettivamente una tensione positiva di 3.3 V e 5 V. Sono utilizzati per fornire alimentazione a vari componenti e moduli collegati alla scheda.
- **GND:** GND rappresenta la tensione di riferimento per il circuito. È essenziale per completare il circuito elettrico, fornendo un percorso di ritorno per la corrente elettrica alla sorgente. In pratica GND è il terminale negativo dell'alimentazione e serve come riferimento comune per tutte le misurazioni di tensione nel circuito.

Pin del microcontrollore (Analogici e Digitali):

- **ANALOG IN:** Questi sono gli ingressi analogici, ovvero i canali dell'ADC interno al microcontrollore. I canali dell'ADC (Analog-to-Digital Converter) sono utilizzati per convertire segnali analogici in segnali digitali che il microcontrollore può elaborare. Arduino Uno dispone di sei canali ADC, numerati da A0 a A5.
- **TX (trasmissione) e RX (ricezione):** Pin utilizzati per la comunicazione seriale.
- **DIGITAL e PWM:** Sono rispettivamente Pin di input/output digitali e uscite PWM.
- **AREF:** Tensione di riferimento analogica utilizzata dall'ADC.
- **Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK):** Pin utilizzati per la comunicazione SPI.
- **Pin ICSP:** Pin utilizzati per la programmazione via ICS del microcontrollore.

Tra i componenti hardware, il microcontrollore con i pin ICS è evidenziato in grigio, il circuito di controllo della tensione, che genera una tensione di 5 V e 3.3 V a partire dalla tensione di ingresso, è evidenziato in blu, mentre il circuito di controllo, comunicazione e programmazione, formato da un microcontrollore ATMEGA16U2, è evidenziato in viola (Figura 7) [4].

2.1.3 Applicazioni di Arduino

Arduino è utilizzato in una vasta gamma di applicazioni, tra cui:

- **Automazione domestica:** Controllo di luci, elettrodomestici e sistemi di sicurezza tramite sensori e attuatori.
- **Robotica:** Creazione di robot autonomi o telecomandati, utilizzando motori, sensori e moduli di comunicazione.
- **Progetti educativi:** Strumento didattico per insegnare elettronica e programmazione in scuole e università.
- **Arte interattiva:** Installazioni artistiche che reagiscono a input ambientali come luce, suono e movimento.

2.2 Sensori PPG: Funzionamento generale

In questo paragrafo viene spiegato il funzionamento dei sensori di frequenza cardiaca, trattati nei paragrafi 2.3, 2.4 e 2.5: sensore Grove – Finger-Clip, Pulse Sensor e sensore MAX30102.

I sensori ottici di frequenza cardiaca operano emettendo una luce verde (lunghezza d'onda pari a 550 nm) o due luci, una rossa e una infrarossa (lunghezze d'onda di 660 nm e 880 nm), sul dito o su un'altra parte del corpo in cui la pelle non sia troppo spessa. La quantità di luce riflessa viene misurata tramite un fotosensore. Questa tecnica di rilevamento ottico degli impulsi è chiamata fotopleetismografia (PPG) ed è utilizzata per rilevare le variazioni di volume del sangue nei vasi sanguigni [8].

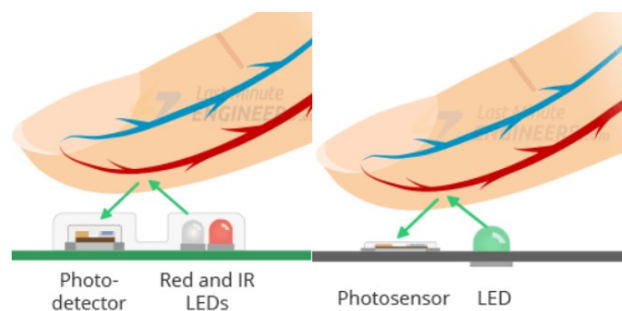


Figura 8: Posizionamento del dito sui sensori ottici di frequenza cardiaca.

L'emoglobina ossigenata nel sangue arterioso ha la capacità di assorbire la luce verde o infrarossa: maggiore è la concentrazione di emoglobina, più intenso sarà l'assorbimento della luce. Ad ogni battito cardiaco, il sangue viene pompato attraverso il dito, provocando una variazione della quantità di luce riflessa. Questa variazione viene rilevata dal fotosensore, generando una forma d'onda: il segnale sarà minimo in corrispondenza del passaggio dell'emoglobina ossigenata (come evidenziato in Figura 9), poiché essa assorbe una maggiore quantità di luce verde o infrarossa.

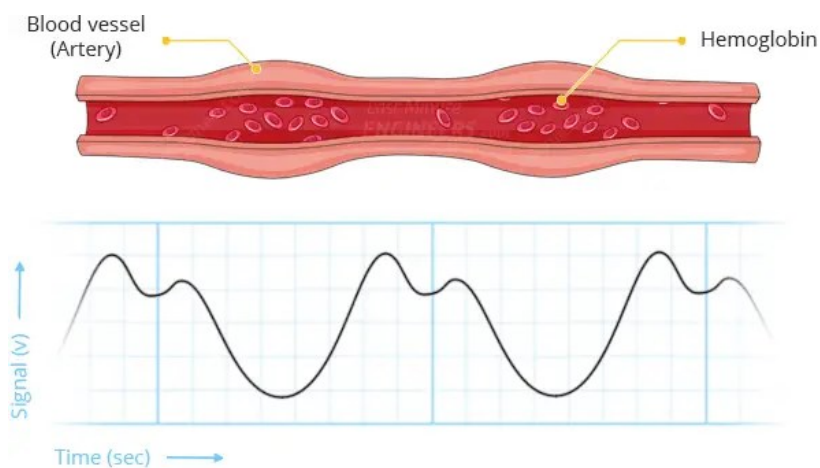


Figura 9: Andamento del segnale nel tempo.

Continuando a illuminare il dito con la luce e a rilevare i dati dal sensore, si può ottenere una misurazione del battito cardiaco.

Il segnale proveniente dal fotosensore è generalmente debole e affetto da rumore. Per questo motivo, solitamente, viene fatto passare attraverso una rete di filtri R/C, che utilizza resistori e condensatori per attenuare il rumore e migliorare la qualità del segnale. Successivamente, il segnale filtrato viene amplificato mediante un amplificatore operazionale. Questo processo permette di ottenere un segnale significativamente più forte, pulito e facilmente rilevabile.

2.3 Sensore di frequenza cardiaca con guscio: Grove – Finger-Clip

Il sensore di frequenza cardiaca Grove - Finger-clip, illustrato in Figura 10, è stato collegato alla scheda Arduino Uno per effettuare le acquisizioni dei dati. Questo sensore utilizza la tecnologia ottica per misurare la variazione del flusso sanguigno nei vasi, permettendo di rilevare con precisione la frequenza cardiaca dell'utente.

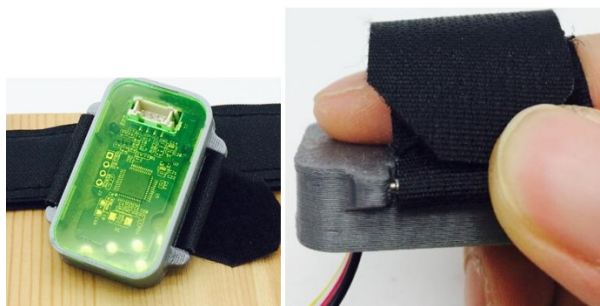


Figura 10: Sensore Grove – Finger-clip.

Il sensore di frequenza cardiaca “**Grove - Finger-clip**” con guscio [5] rappresenta un avanzamento significativo nella tecnologia dei dispositivi indossabili per il monitoraggio della salute. È basato sul PAH8001EI-2G, un sensore ottico ad alte prestazioni e a basso consumo energetico, dotato di LED verde e DSP¹ integrato per il rilevamento della frequenza cardiaca (HRD).

Caratteristiche tecniche: Il sensore è progettato per essere altamente efficiente dal punto di vista energetico, con modalità di risparmio energetico flessibili che lo rendono ideale per l'uso in dispositivi indossabili. Il basso consumo energetico è un fattore cruciale per garantire una lunga durata della batteria nei dispositivi portatili, migliorando così l'esperienza dell'utente.

Integrazione con STM32²: Un aspetto distintivo di questo modulo è l'integrazione di un microcontrollore STM32, che gestisce l'elaborazione dei dati della frequenza

¹ **DSP** sta per **Digital Signal Processor**. È un tipo di microprocessore specializzato nell'elaborazione dei segnali digitali in tempo reale. Nel contesto del sensore di frequenza cardiaca, il DSP è utilizzato per elaborare i dati raccolti dal sensore ottico e calcolare la frequenza cardiaca in modo accurato e rapido.

² STM32 è una famiglia di microcontrollori sviluppata da STMicroelectronics, basata su core ARM Cortex-M. Questi microcontrollori sono progettati per offrire alte prestazioni, basso consumo energetico, capacità di elaborazione in tempo reale e connettività avanzata, mantenendo al contempo un'integrazione completa e facilità di sviluppo.

cardiaca. L'interfaccia SWD³ consente agli utenti di riprogrammare il microcontrollore, offrendo una notevole flessibilità per personalizzare e ottimizzare le funzionalità del sensore in base alle specifiche esigenze applicative.

Design e usabilità: Il sensore è dotato di un guscio protettivo realizzato con stampa 3D e di due fasce, che permettono di fissarlo facilmente al dito, al polso o al braccio. Questo design ergonomico garantisce un buon contatto con la pelle, essenziale per ottenere misurazioni accurate della frequenza cardiaca. La facilità di fissaggio e la stabilità del sensore durante l'uso sono fondamentali per ridurre al minimo gli errori di misurazione dovuti a movimenti o contatti non ottimali.

Applicazioni e implicazioni future: Grazie alle sue caratteristiche avanzate, il sensore di frequenza cardiaca Grove - Finger-clip con guscio ha un ampio spettro di applicazioni, che vanno dal monitoraggio della salute personale alla ricerca medica. La possibilità di integrare questo sensore in vari dispositivi indossabili apre nuove prospettive per il monitoraggio continuo e non invasivo della salute, contribuendo a migliorare la qualità della vita degli utenti.

Le caratteristiche tecniche avanzate, combinate con un design ergonomico e una facile integrazione con sistemi di microcontrollori, lo rendono un componente essenziale per lo sviluppo di dispositivi indossabili di nuova generazione.

Riassumendo, il sensore Grove – Finger-clip presenta le seguenti specifiche tecniche:

- Bassissimo consumo energetico, modalità di risparmio energetico durante il tempo di non movimento tattile
- Controllo flessibile della frequenza in modalità sleep
- Integrato il STM32F103
- Interfaccia I2C
- Area del sensore di frequenza cardiaca di soli 3,0 x 4,7 mm²
- Interfaccia SWD riservata
- Va equipaggiato con conchiglia e fasce
- Temperatura di lavoro: da -20 a +60°C

³ L'interfaccia SWD (Serial Wire Debug) è un protocollo di debug e programmazione utilizzato per i microcontrollori ARM, come l'STM32. L'interfaccia SWD riservata nel modulo del sensore di frequenza cardiaca offre una grande flessibilità e facilità d'uso, permettendo agli sviluppatori di personalizzare e ottimizzare il microcontrollore STM32 per ottenere le migliori prestazioni possibili dal sensore.

- Il connettore **nero** collega il pin **GND** del sensore al corrispondente della scheda. GND o Ground è il PIN della tensione di massa e rappresenta il punto di riferimento comune nei circuiti elettrici.
- Il connettore **rosso** viene collegato al pin della tensione di alimentazione (**VCC**), che nel nostro caso è **3.3 V**.
- Il connettore **bianco** collega il pin **SDA** (Serial Data) del sensore al pin analogico **A4**. Il pin SDA è il canale attraverso cui i dati vengono inviati e ricevuti.
- Il connettore **giallo** collega il pin **SCL** (Serial Clock) del sensore al pin analogico **A5**. Il pin SCL è utilizzato per mantenere sincronizzati i diversi dispositivi tramite un segnale di clock [6].

2.3.1.2 Secondo collegamento del sensore Grove-Finger-Clip al microcontrollore: Pin Digitali

Il sensore Grove-Finger-Clip va collegato ad Arduino UNO come illustrato in Figura 13:

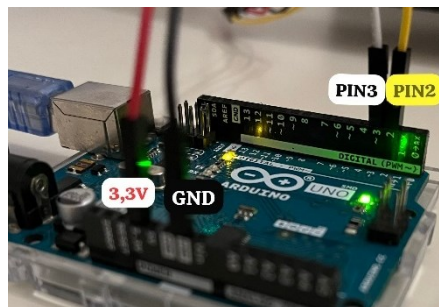


Figura 13: Secondo collegamento del sensore Grove-Finger-Clip alla scheda.

- Il connettore **nero** collega il pin **GND** del sensore al corrispondente della scheda.
- Il connettore **rosso** viene collegato al pin della tensione di alimentazione (**VCC**), che nel nostro caso è **3.3 V**.
- Il connettore **bianco** collega l'SDA (Serial Data) del sensore al pin **digitale 3**.
- Il connettore **giallo** collega il pin SCL (Serial Clock) del sensore al pin **digitale 2**.

2.4 Pulse Sensor

Pulse sensor è un sensore open source di frequenza cardiaca a basso consumo energetico, progettato per essere utilizzato con Arduino [7]. In questo progetto di tesi il Pulse sensor, illustrato in Figura 14, è stato collegato alla scheda Arduino Uno per effettuare le acquisizioni dei dati.



Figura 14: Pulse sensor.

2.4.1 Funzionamento hardware

Il dito va posizionato nella parte inferiore del sensore, dove si trova il logo del cuore, al centro del quale vi è una piccola apertura circolare, attraverso la quale passa un fascio di luce verde generato da un LED [7]. Appena sotto l'apertura si trova un piccolo sensore di luce ambientale: APDS-9008 di Avago.

Sul retro del sensore sono presenti un amplificatore operazionale MCP6001 di Microchip e vari resistori e condensatori che formano la rete di filtri R/C. Per prevenire i danni in caso di collegamento errato dei cavi di alimentazione è stato aggiunto un diodo di protezione contro l'inversione di polarità. Il modulo, per funzionare correttamente, necessita in una tensione di alimentazione continua compresa tra 3.3 e 5 V. Inoltre, il sensore consuma meno di 4 milliamper (mA) di corrente.

Nella Figura 15 sono indicate le posizioni dei principali componenti del sensore su entrambe le facce.

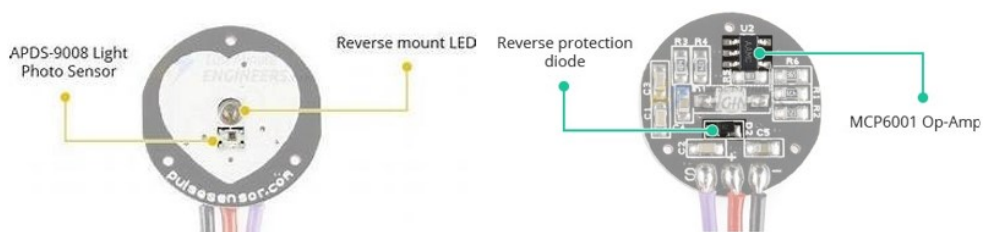


Figura 15: Componenti del Pulse Sensor.

Nella Figura 16 sono elencate le specifiche tecniche del Pulse sensor:

	VCC	3,0 – 5,5 V
Valutazioni massime	IMax (Assorbimento di corrente massimo)	< 4mA
	VOut (Intervallo di tensione di uscita)	Da 0,3 V a VCC
Lunghezza d'onda	Uscita LED	565 nm
	Ingresso sensore	525 nm
Dimensioni	L x P (PCB)	15,8 mm (0,625")
	Lunghezza del cavo	20 cm (7,8")

Figura 16: Specifiche tecniche del Pulse sensor.

2.4.2 Collegamento del Pulse Sensor all'Arduino Uno

Il Pulse Sensor ha tre connettori maschio header: sono piccoli pin che possono essere inseriti in connettori femmina su una scheda, nel nostro caso l'Arduino UNO, per stabilire una connessione elettrica.

Il sensore deve essere collegato alla scheda Arduino Uno mediante tre jumpers, come illustrato in Figura 17:

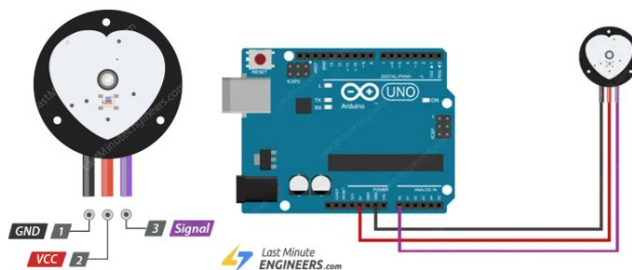


Figura 17: Piedinatura e cablaggio del Pulse Sensor.

I collegamenti da effettuare sono i seguenti:

- Il cavo **nero** collega il PIN **GND** (-) del sensore al perno di messa a terra di Arduino.
- Il cavo **rosso** collega il PIN **VCC** del sensore al PIN da 3.3 o 5 V di Arduino, che corrisponde alla tensione di alimentazione, che nel nostro caso è di **3.3 V**.
- Il cavo **viola** collega il PIN dell'uscita del segnale (S) del sensore all'ingresso analogico **A0** di Arduino.

2.5 Sensore MAX30102

Il sensore di frequenza cardiaca MAX30102 è un dispositivo biometrico plug-and-play a bassa potenza, basato sull'interfaccia I2C, utilizzato per misurare la frequenza cardiaca e la saturazione di ossigeno nel sangue (SpO2). Questo sensore è ideale per studenti, hobbisti, ingegneri, produttori e sviluppatori di giochi e dispositivi mobili che intendono integrare dati in tempo reale sulla frequenza cardiaca nei loro progetti [14].

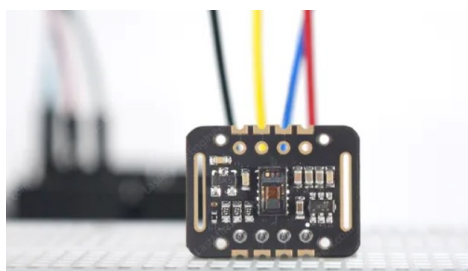


Figura 18: Sensore di frequenza cardiaca MAX30102.

Funzionamento del Sensore:

Questo sensore integra un pulsossimetro e un misuratore di frequenza cardiaca in un unico circuito integrato (IC). Il MAX30102 combina due LED (uno rosso e uno infrarosso), un fotorilevatore, un'ottica ottimizzata e un'elaborazione del segnale analogico a basso rumore per rilevare i segnali di pulsossimetria (SpO2) e frequenza cardiaca (HR).

Come illustrato in Figura 19, a fianco dei due LED si trova un fotorilevatore altamente sensibile. Il principio di funzionamento prevede l'illuminazione alternata dei LED, con il rilevamento della quantità di luce riflessa verso il fotorilevatore. Analizzando il tracciato della luce riflessa, è possibile misurare il livello di ossigeno nel sangue e la frequenza cardiaca.

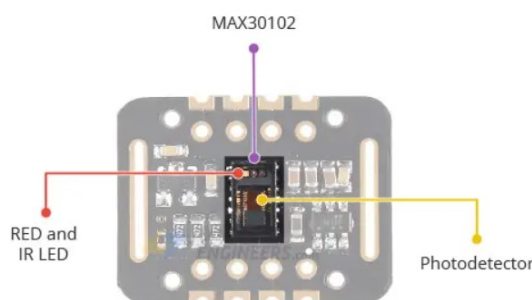


Figura 19: Panoramica dell'hardware del sensore MAX30102.

Il sensore MAX30102, a differenza dei sensori Grove-Finger-Clip e Pulse Sensor (descritti nei paragrafi 2.3 e 2.4), che utilizzano un solo LED verde, è dotato di due LED: uno rosso e uno infrarosso. Questo permette al sensore di misurare non solo la frequenza cardiaca, come descritto nel paragrafo 2.2, ma anche il livello di ossigeno nel sangue. Questa tecnica, chiamata “pulsossimetria”, si basa sul principio che la quantità di luce rossa e infrarossa assorbita varia in base alla quantità di ossigeno nel sangue.

Il grafico di Figura 20 mostra lo spettro di assorbimento dell'emoglobina ossigenata (HbO₂) e dell'emoglobina deossigenata (Hb). Come si può osservare, il sangue deossigenato assorbe più luce rossa (660 nm), mentre il sangue ossigenato assorbe maggiormente la luce infrarossa (880 nm). Misurando il rapporto tra la luce infrarossa e quella rossa ricevuta dal fotorilevatore, è possibile calcolare il livello di ossigeno nel sangue (SpO₂).

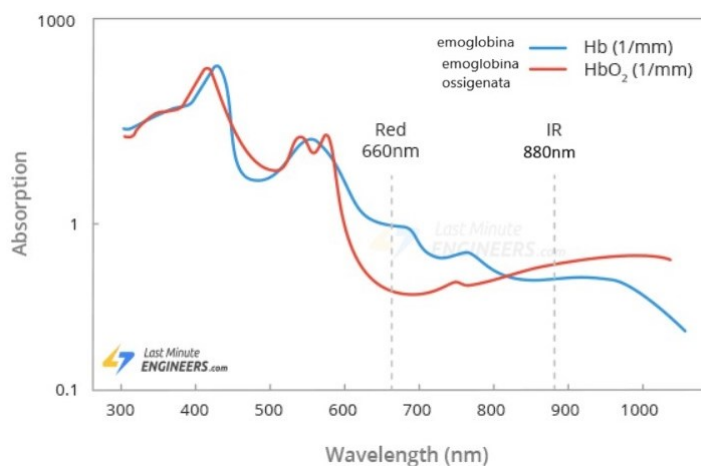


Figura 20: Spettro di assorbimento dell'emoglobina ossigenata e deossigenata.

Caratteristiche Tecniche del Sensore:

Il chip MAX30102 richiede due diverse tensioni di alimentazione: 1.8 V per l'IC e 3.3 V per i LED rossi e infrarossi. Pertanto, il modulo è dotato di regolatori di tensione da 3.3V e 1.8 V, come illustrato in Figura 21.

Sul retro del PCB (scheda a circuito stampato, Figura 21) è presente un ponticello a saldare che consente di selezionare tra il livello logico di 3.3 V e quello di 1.8 V. Di default, è impostato su 3.3 V, compatibile con i livelli logici di Arduino, ma può essere

modificato a 1.8 V secondo le necessità. Questo permette di collegare il modulo al microcontrollore con livelli I/O di 5 V, 3.3 V e anche 1.8 V.

Una delle caratteristiche più rilevanti del MAX30102 è il suo basso consumo energetico: durante la misurazione, consuma meno di 600 μA . Inoltre, può essere messo in modalità standby, dove il consumo scende a soli 0.7 μA . Questo basso consumo energetico rende il MAX30102 ideale per l'uso in dispositivi alimentati a batteria, come telefoni, dispositivi indossabili o orologi intelligenti.

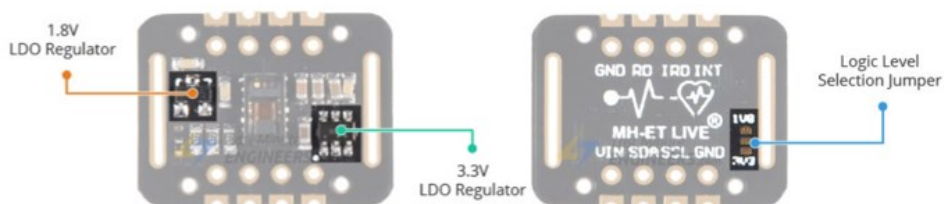


Figura 21: Vista anteriore e posteriore della scheda a circuito stampato (PCB) del sensore MAX30102.

Il MAX30102 è equipaggiato di un sensore di temperatura integrato, utilizzabile per compensare le variazioni ambientali e calibrare le misurazioni. Questo sensore di temperatura, che misura la temperatura del chip del sensore MAX30102, opera in un intervallo tra $-40\text{ }^{\circ}\text{C}$ a $+85\text{ }^{\circ}\text{C}$ con una precisione di $\pm 1\text{ }^{\circ}\text{C}$.

Interfaccia e Programmazione:

Il sensore MAX30102 utilizza un'interfaccia I2C a due fili per la comunicazione con il microcontrollore, rendendolo semplice da integrare in vari progetti. L'indirizzo I2C del sensore è fisso: 0xAE per le operazioni di scrittura e 0xAF per le operazioni di lettura. Il sensore è dotato di un buffer FIFO (First-In-First-Out) per l'archiviazione dei campioni di dati, con la capacità di memorizzare fino a 32 campioni di dati (SpO2 e frequenza cardiaca). Questo riduce il carico di lavoro del microcontrollore, ottimizzando il consumo energetico.

Inoltre, il MAX30102 può generare interruzioni programmabili per notificare al microcontrollore eventi specifici, come la disponibilità di nuovi dati o condizioni di errore. Questo permette al microcontrollore di eseguire altre operazioni mentre il sensore raccoglie i dati, migliorando l'efficienza complessiva del sistema. La programmazione

del sensore può essere effettuata utilizzando librerie disponibili per piattaforme come Arduino, facilitando l'implementazione e la gestione dei dati raccolti.

Specifiche Tecniche del sensore MAX30102 (Figura 22):

Alimentatore	Da 3,3 V a 5,5 V
Assorbimento di corrente	~600µA (durante le misurazioni)
	~0,7 µA (in modalità standby)
Lunghezza d'onda del LED rosso	660 nm
Lunghezza d'onda del LED IR	880 nm
Intervallo di temperatura	Da -40 °C a +85 °C
Precisione della temperatura	±1°C

Figura 22: Specifiche Tecniche del sensore MAX30102.

2.5.1 Collegamento del sensore MAX30102 ad Arduino Uno

Prima di illustrare come collegare il sensore MAX30102 ad Arduino, è necessario presentare in Figura 23 la piedinatura del sensore stesso.

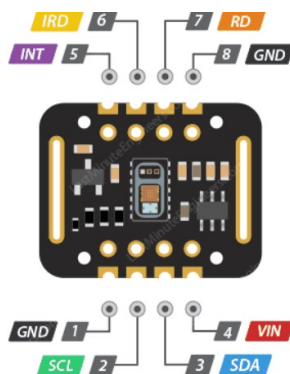


Figura 23: Piedinatura del sensore MAX30102.

VIN: è il PIN di alimentazione. Va collegato all'uscita da 3.3 V o 5 V di Arduino.

SCL: è il PIN di clock, va collegato alla linea di clock I2C di Arduino.

SDA: è il PIN che va collegato alla linea dei dati I2C di Arduino.

INT: è un PIN di interrupt che può essere programmato per segnalare eventi specifici, come la rilevazione di un impulso. Questo pin è configurato come drain aperto, il che

significa che necessita di un resistore di pull-up per mantenere il livello logico alto quando non è attivo. Quando si verifica un evento di interrupt, il pin INT passa a livello basso (LOW) e rimane in questo stato fino a quando l'interrupt non viene cancellato.

IRD: è il PIN che controlla il LED infrarosso (IR). Questo pin è utilizzato per accendere e spegnere il LED infrarosso durante le misurazioni.

RD: è simile al pin IRD, ma viene utilizzato per pilotare il LED rosso.

GND: è il PIN di massa o terra.

In Figura 24 è mostrato l'opportuno cablaggio tra il sensore MAX30102 e la scheda Arduino UNO suggerito per l'utilizzo dei codici della libreria "SparkFun Electronics" di Arduino [14].

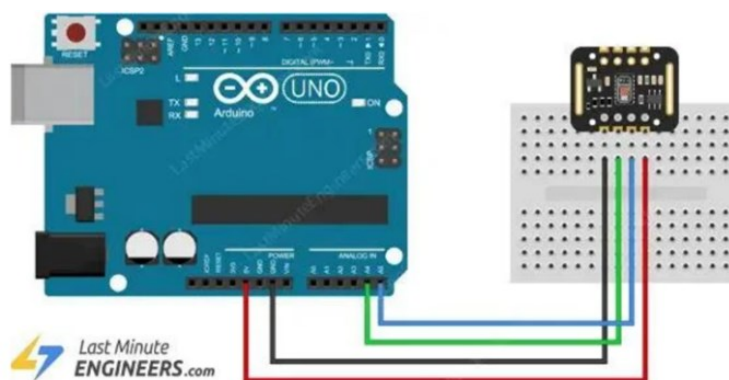


Figura 24: Cablaggio tra il sensore MAX30102 e la scheda Arduino UNO.

Il pin VCC del sensore deve essere collegato all'alimentazione, che può essere di 5 V o 3.3 V. È fondamentale assicurarsi che la tensione utilizzata sia la stessa della logica del microcontrollore. Nella maggior parte delle schede Arduino, questa tensione è di 5 V. Tuttavia, per i dispositivi con logica a 3.3 V, è necessario utilizzare 3.3 V. Successivamente, il pin GND deve essere collegato al pin di messa a terra della scheda. Il pin SCL deve essere collegato al pin del clock I2C, mentre il pin SDA deve essere collegato al pin dati dell'Arduino. Ogni scheda Arduino ha pin differenti per queste connessioni, che devono essere collegati di conseguenza. Sulla scheda Arduino UNO, i pin SDA (linea dati) e SCL (linea di clock) si trovano sulle intestazioni dei pin vicino al pin AREF. Questi pin sono anche noti come A5 (SCL) e A4 (SDA).

2.6 Il Pulsossimetro Beurer

Il pulsossimetro “Beurer” è un dispositivo medico acquistabile in farmacia per monitorare la saturazione di ossigeno nel sangue (SpO_2) e la frequenza cardiaca (BPM) in ambito domestico.



Figura 25: Pulsossimetro Beurer.

In questo progetto di tesi l’uso del pulsossimetro è stato fondamentale per verificare l’accuratezza delle misurazioni effettuate con gli altri sensori.

Durante i test, descritti nel dettaglio nel Capitolo 3, il soggetto indossava il sensore da implementare su una mano e il pulsossimetro sull’altra. Questo approccio ha permesso di confrontare direttamente le misurazioni ottenute, garantendo una valutazione precisa e affidabile delle prestazioni dei sensori analizzati. Sebbene le acquisizioni effettuate dal pulsossimetro non siano state incluse nella tesi, i dati raccolti sono stati verificati visivamente durante le prove. Questa verifica visiva ha premesso di stabilire se i dati del pulsossimetro coincidevano o meno con le acquisizioni degli altri sensori, evidenziando così la presenza di dati corrotti, limitazioni o problematiche.

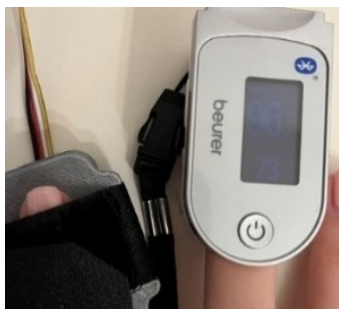


Figura 26: Sensore e pulsossimetro indossati.

Il principio di funzionamento del pulsossimetro differisce da quello dei sensori utilizzati. Il pulsossimetro emette due tipi di luce, una rossa e una infrarossa, e misura la quantità di luce trasmessa attraverso il dito. Utilizza queste informazioni per calcolare la saturazione di ossigeno nel sangue e la frequenza cardiaca [13]. Al contrario, i tre sensori esaminati misurano la quantità di luce riflessa dal dito, e non quella trasmessa.

Capitolo 3: Implementazione e Validazione dei codici per l'acquisizione della Frequenza Cardiaca

In questo capitolo vengono presentati i vari algoritmi sviluppati durante lo studio e le prove sperimentali condotte per verificarne l'efficacia. Sono stati realizzati cinque algoritmi: il primo è un algoritmo in MATLAB per l'analisi statistica, che verrà trattata successivamente; il secondo e il terzo sono due codici in Arduino per leggere i dati della frequenza cardiaca dal sensore Grove – Finger-clip; il quarto e il quinto sono due codici in Arduino per monitorare i battiti cardiaci utilizzando il sensore Pulse Sensor, uno per le acquisizioni tramite Coolterm e l'altro con il visualizzatore Processing. Infine, nell'ultimo paragrafo di questo capitolo, vengono illustrati i diversi collegamenti e i codici eseguiti per il funzionamento del sensore MAX30102.

3.1 Implementazione dell'algoritmo MATLAB per l'analisi statistica delle acquisizioni di frequenza cardiaca

In questo paragrafo viene descritto il codice MATLAB, riportato in Figura 27, realizzato per caricare, elaborare e visualizzare i dati di frequenza cardiaca raccolti durante le prove sperimentali.

MATLAB è un ambiente di programmazione estremamente semplice e veloce, che permette di accedere ad una vasta libreria di procedure di calcolo numerico avanzate, senza la necessità di implementarle manualmente. È ampiamente utilizzato nella ricerca scientifica perché mette a disposizione molti toolboxes per l'analisi dei segnali e la simulazione di sistemi complessi, includendo anche librerie per l'interfacciamento con hardware. In questo lavoro di tesi è stata utilizzata solo una piccola parte delle funzionalità offerte da MATLAB [9].

```
3 %rimuove tutte le variabili dell'ambiente di lavoro di MATLAB, liberando la
4 % memoria
5 clear all
6 %Chiude tutte le finestre delle figure aperte
7 close all
8 %Cancella la finestra del comando, pulendo lo schermo
9 clc
10
11 %%
12 %Carica i dati del file "acq2.mat", che contiene la matrice acq2
13 load("acq2.mat");
14
15 %Estrae la prima colonna della matrice "acq2" e la assegna alla variabile
16 %bpm
17 bpm = acq2(:,1);
18
19 %Calcola il numero totale di elementi della colonna bpm
20 n1 = numel(bpm);
21
22 %Determina il tempo totale in secondi
23 n2=n1/2;
24
25 %Crea una colonna del vettore dei tempi che va da 0,5 a n2
26 sec = [0.5:0.5:n2]';
27
28 %crea una nuova finestra per il grafico
29 figure;
30
31 %traccia il grafico della frequenza cardiaca in funzione del tempo
32 plot(sec,bpm);
33
34 %nomina gli assi
35 xlabel('sec');
36 ylabel('bpm');
```

Figura 27: Codice MATLAB per creare il grafico Tempo-Frequenza cardiaca [s-bpm].

Descrizione del codice:

Il codice MATLAB carica un file di dati, estrae la frequenza cardiaca e la visualizza in un grafico.

- Riga 13, **load ("acq2.mat");** : Carica il file "acq2.mat" nel workspace di MATLAB. Il file contiene una matrice di dati, dove ogni colonna rappresenta un diverso set di dati acquisiti. In questo caso il file è costituito solo da una colonna.

- Riga 17, **bpm = acq2 (:, 1);** : Estrae la prima colonna della matrice “acq2” e la assegna alla variabile “bpm”. Questa colonna contiene i dati della frequenza cardiaca.
- Riga 20, **n1 = numel (bpm);** : n1 è il numero totale di elementi del vettore “bpm”.
- Riga 23, **n2 = n1/2;** : n2 è il tempo in cui sono stati acquisiti i dati, viene calcolato come la metà del numero totale di elementi perché i dati sono acquisiti ogni 0,5 secondi.
- Riga 26, **sec = [0.5 : 0.5 : n2]’;** : Questa riga crea il vettore del tempo in secondi. Il vettore inizia a 0,5 secondi e incrementa di 0,5 secondi fino a “n2” secondi. Il simbolo, ‘, trasforma il vettore riga in un vettore colonna.
- Righe 28-36: Creano un grafico della frequenza cardiaca “bpm” in funzione del tempo “sec”. Nomina gli assi X e Y.

Questo codice permette di visualizzare come varia la frequenza cardiaca nel tempo, fornendo un’analisi visiva dei dati acquisiti.

3.2 Implementazione dei codici per Grove - Finger-Clip

In questo paragrafo vengono illustrati i due codici sviluppati per la lettura della frequenza cardiaca utilizzando il sensore Grove – Finger-Clip collegato tramite il protocollo I2C ⁴. I codici sono stati scritti nel linguaggio di programmazione C++ e caricati su una scheda Arduino. Vengono spiegate le varie parti dei codici, il loro funzionamento e come queste contribuiscano al rilevamento e alla visualizzazione dei dati della frequenza cardiaca.

I due codici operano in modo simile, differenziandosi unicamente per il metodo di collegamento del sensore al microcontrollore: il primo codice utilizza gli ingressi analogici A4 e A5 per i segnali di dati e clock (come mostrato nel paragrafo 2.3.1.1), mentre il secondo codice permette il collegamento tramite i PIN digitali 2 e 3 (paragrafo 2.3.1.2).

⁴ Il protocollo I2C (Inter-Integrated Circuit) è un modo semplice per far comunicare tra loro diversi dispositivi elettronici utilizzando solo due fili: uno per i dati (SDA) e uno per il clock (SCL) [11].

3.2.1 Primo codice per il Grove-Finger-Clip: Collegamento tramite i PIN analogici

Il primo codice implementato in Arduino è mostrato in Figura 28 [10]:

```
6  #include <Wire.h>
7  void setup() {
8      Serial.begin(9600);
9      Serial.println("heart rate sensor:");
10     Wire.begin();
11 }
12
13 void loop() {
14     Wire.requestFrom(0x50, 1);    // request 1 bytes from slave device
15     while(Wire.available()) {    // slave may send less than requested
16         unsigned char c = Wire.read(); // receive heart rate value (a byte)
17         Serial.println(c, DEC);    // print heart rate value
18     }
19     delay(500);
20 }
```

Figura 28: Primo codice per il sensore Grove-Finger-Clip.

Descrizione del codice:

- Riga 6: La libreria “**Wire.h**” è inclusa per abilitare la comunicazione I2C tra l’Arduino e il sensore di frequenza cardiaca. Questa libreria fornisce le informazioni necessarie per inizializzare la comunicazione e scambiare dati con dispositivi I2C.
- Le righe 7-11 fanno parte della funzione “**setup ()**”, che viene eseguita una sola volta all’avvio del programma. In questa funzione troviamo tre comandi principali:
 - **Serial.begin(9600)**: Inizializza la comunicazione seriale a 9600 baud, permettendo ad Arduino di inviare dati al computer.
 - **Serial.println(“heart rate sensor:”)**: Stampa un messaggio di avvio sul monitor seriale per indicare che il programma è iniziato.
 - **Wire.begin()**: Inizializza la libreria Wire.
- Le righe 13-20 fanno parte della funzione “**loop ()**”, che viene eseguita ripetutamente per mantenere il programma in esecuzione continua. In questa funzione troviamo:
 - **Wire.requestFrom(0x50,1)**: richiede 1 byte di dati dal sensore con indirizzo I2C 0x50. Per questo comando è stato necessario

utilizzare un codice che ha permesso di stabilire l'indirizzamento corretto tra il sensore e la scheda.

- **while(Wire.available()):** Controlla se ci sono dati disponibili dal sensore.
- **unsigned char c = Wire.read():** Legge un byte di dati dal sensore, che rappresenta il valore della frequenza cardiaca.
- **Serial.println(c, DEC):** Stampa il valore della frequenza cardiaca in formato decimale sul monitor seriale.
- **Delay (500):** Aspetta 500 millisecondi prima di ripetere il ciclo.

Dopo aver collegato il sensore al microcontrollore, come illustrato nel paragrafo 2.3.1.1, il codice consente di leggere i dati della frequenza cardiaca dal sensore Grove – Finger-Clip tramite il protocollo I2C e di visualizzarli sul monitor seriale di Arduino, nel nostro caso, abbiamo utilizzato il programma Coolterm [12] che permette di salvare i dati acquisiti da Arduino in un file di testo .txt.

3.2.2 Secondo codice per il Grove-Finger-Clip: Collegamento tramite i PIN digitali

Il secondo codice implementato in Arduino è mostrato in Figura 29:

```
1  #define SDA_PORT PORTD
2  #define SDA_PIN 3
3  #define SCL_PORT PORTD
4  #define SCL_PIN 2
5  #include <SoftI2CMaster.h>
6  #include <SoftWire.h>
7
8  #define SoftWire Wire
9
10
11 void setup() {
12     Serial.begin(9600);
13     Serial.println("heart rate sensor:");
14     SoftWire.begin();
15 }
16 void loop() {
17     SoftWire.requestFrom(0x50, 1); // request 1 bytes from slave device
18     while(SoftWire.available()) { // slave may send less than requested
19         unsigned char c = SoftWire.read(); // receive heart rate value (a byte)
20         Serial.println(c, DEC); // print heart rate value
21     }
22     delay(500);
23 }
```

Figura 29: Secondo codice per il sensore Grove-Finger-Clip.

- Righe 1-8: In queste righe vengono definiti i PIN utilizzati per il collegamento I2C (SDA e SCL) e vengono incluse le librerie necessarie per la comunicazione I2C software.
- Righe 11-15: La funzione “*setup*” inizializza la comunicazione seriale a 9600 baud per la visualizzazione dei dati e avvia la comunicazione I2C.
- Righe 16-23: La funzione “*loop*” richiede un byte di dati dal dispositivo con indirizzo 0x50, che rappresenta il valore della frequenza cardiaca. I dati ricevuti vengono letti e stampati sulla seriale. La funzione “*delay(500)*” introduce una pausa di 500 millisecondi tra le letture successive.

3.2.3 Prove sperimentali per verificare l’efficacia dell’algoritmo

Una volta completata l’implementazione del codice su Arduino UNO, sono stati eseguiti dei test per valutarne l’efficacia e identificare eventuali problematiche. Per questi test è stato utilizzato il primo codice descritto nel paragrafo 3.2.1, sebbene l’utilizzo del secondo codice descritto nel paragrafo 3.2.2 avrebbe prodotto gli stessi risultati.

Il sensore Grove – Finger-Clip è stato collegato alla scheda come descritto nella sezione 2.3.1.1, e la scheda è stata collegata al computer tramite l’apposito ingresso USB.

Le prove sono state condotte su quattro soggetti: due donne di 22 e 47 anni e due uomini di 12 e 25 anni. Il sensore è stato posizionato saldamente sul dito indice della mano dominante di ciascun soggetto, assicurandosi che l’area del sensore fosse a contatto con la pelle.

Durante tutti i test, i soggetti indossavano il sensore Grove – Finger-Clip per acquisire i battiti cardiaci, visualizzati su Coolterm. Contemporaneamente, sull’altra mano, tenevano il pulsossimetro “Beurer”, un dispositivo medico acquistabile in farmacia per monitorare la frequenza cardiaca in ambito domestico. L’utilizzo di quest’ultimo è stato fondamentale per confermare l’accuratezza delle misurazioni effettuate dal sensore.

Le due donne sono state sottoposte a due test: uno a riposo e uno in condizioni di affaticamento. Mentre, sugli uomini è stato eseguito il test solo in condizioni di riposo.

In tutti i casi, le misurazioni eseguite dal sensore Grove – Finger-Clip sono risultate coincidenti con quelle ottenute tramite il pulsossimetro. Le acquisizioni del pulsossimetro sono state verificate visivamente, mentre i dati del sensore Grove-Finger-Clip sono stati salvati su CoolTerm e successivamente elaborati in grafici MATLAB (come mostrato nelle figure seguenti). Pertanto, possiamo affermare che il codice sviluppato è corretto e che il sensore funziona adeguatamente.

Di seguito sono riportati i grafici delle acquisizioni della frequenza cardiaca ottenuti con il sensore Grove-Finger-Clip. Questi grafici sono stati realizzati utilizzando il codice MATLAB illustrato in Figura 27 e si riferiscono alle prove condotte sui quattro soggetti. Nelle Figura 30 e Figura 31, è illustrato l'andamento dei battiti cardiaci nel tempo per le due donne dopo la corsa. È interessante notare come la frequenza cardiaca sia inizialmente elevata e diminuisca gradualmente durante la fase di recupero, fino a stabilizzarsi.

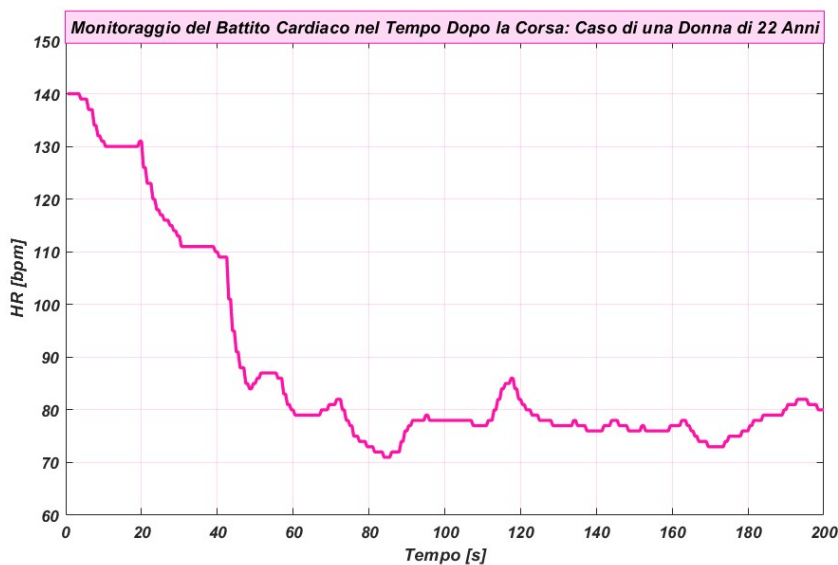


Figura 30: Acquisizioni con il sensore Grove-Finger-Clip su donna di 22 anni dopo la corsa.

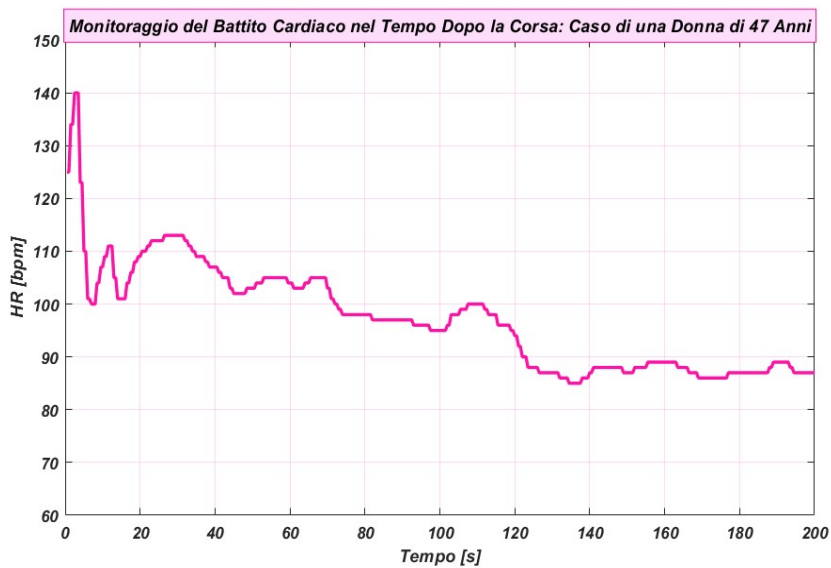


Figura 31: Acquisizioni con il sensore Grove-Finger-Clip su donna di 47 anni dopo la corsa.

Nella Figura 32 viene mostrato l'andamento della frequenza cardiaca nel tempo per un ragazzo di 25 anni in condizioni di riposo. La frequenza cardiaca è costante e si mantiene attorno ai 77 BPM. I grafici relativi agli altri soggetti non sono riportati poiché presentano andamenti molto simili.

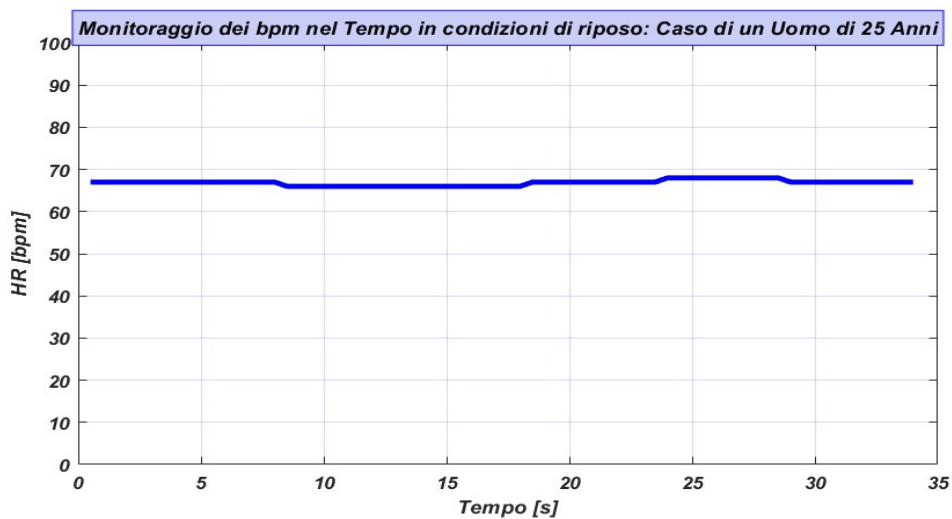


Figura 32: Acquisizioni con il sensore Grove-Finger-Clip su un uomo di 25 anni a riposo.

Per ottenere delle acquisizioni accurate con il sensore Grove-Finger-Clip, è fondamentale fissare saldamente il dispositivo, garantire uno stretto contatto con la pelle ed evitare

movimenti bruschi. È stato osservato che anche il minimo movimento può compromettere la precisione della misurazione della frequenza cardiaca.

La Figura 33 mostra un grafico delle acquisizioni eseguite in condizioni non ottimali: il soggetto si muoveva e il sensore Grove-Finger-Clip non era sempre a stretto contatto con la pelle. Nella parte iniziale del grafico si nota che il sensore impiega qualche secondo per iniziare a misurare correttamente. Questo comportamento è stato riscontrato in tutte le acquisizioni. Nei tre grafici precedenti, la parte iniziale è stata omessa perché avvenuta in condizioni non ottimali, producendo dati palesemente corrotti e inutilizzabili. Questa scelta consente una lettura più rapida e chiara dei dati.

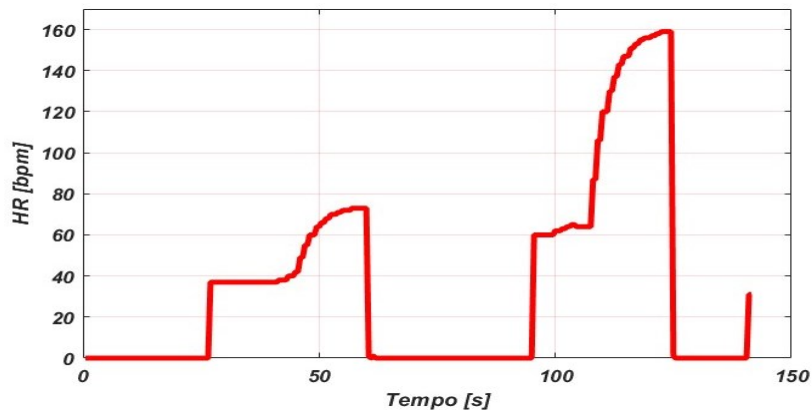


Figura 33: Acquisizione su soggetto in condizioni non ottimali con il sensore Grove-Finger-Clip.

Dal grafico è evidente che il soggetto, compiendo movimenti come alzarsi dalla sedia, gesticolare e muovere le mani, non ha garantito il corretto posizionamento del sensore. Di conseguenza, il sensore ha misurato dati improbabili per le condizioni di riposo in cui si trovava il soggetto.

3.3 Implementazione dei codici per Pulse Sensor

In questo paragrafo vengono descritti i tre codici sviluppati per la lettura della frequenza cardiaca utilizzando il sensore Pulse Sensor [7].

Il primo codice permette di sincronizzare il lampeggiamento del LED integrato di Arduino con il battito cardiaco.

Il secondo codice misura i battiti cardiaci e permette di visualizzarli su CoolTerm.

Il terzo codice è indispensabile per l'utilizzo del software "Processing Visualizer", che mostra il tracciato della frequenza cardiaca, i BPM e l'IBI (Intervallo tra i battiti).

3.3.1 Primo Codice: Tracciamento del battito cardiaco

In questo paragrafo viene descritto lo sketch che fa lampeggiare il LED di Arduino con la stessa frequenza del battito cardiaco. Il codice, denominato "GettingStartedProject" in Figura 34, è disponibile tra gli sketch di esempio nell'IDE di Arduino.

```
1 // Variables
2 int PulseSensorPurplePin = 0; // Pulse Sensor PURPLE WIRE connected to ANALOG PIN 0
3 int LED13 = 13; // The on-board Arduino LED
4
5 int Signal; // holds the incoming raw data. Signal value can range from 0-1024
6 int Threshold = 335; // Determine which Signal to "count as a beat", and which to ignore.
7
8
9 // The Setup Function:
10 void setup() {
11   pinMode(LED13,OUTPUT); // pin that will blink to your heartbeat!
12   Serial.begin(9600); // Set's up Serial Communication at certain speed.
13 }
14
15
16 // The Main Loop Function
17 void loop() {
18
19   Signal = analogRead(PulseSensorPurplePin); // Read the PulseSensor's value.
20   // Assign this value to the "Signal" variable.
21
22   Serial.println(Signal); // Send the Signal value to Serial Plotter.
23
24
25   if(Signal > Threshold){ // If the signal is above "550", then "turn-on" Arduino's on-Board LED.
26     digitalWrite(LED13,HIGH);
27   } else {
28     digitalWrite(LED13,LOW); // Else, the signal must be below "550", so "turn-off" this LED.
29   }
30
31   delay(20);
32 }
33 }
```

Figura 34: Primo codice per il Pulse Sensor.

Spiegazione del codice:

- Righe 1-6: Dichiarazione delle variabili
 - **int PulseSensorPurplePin = 0;** : Il filo che trasferisce il segnale del sensore è collegato al PIN analogico A0.
 - **int LED13 = 13;** : Definisce il LED integrato sulla scheda Arduino.
 - **int Signal;** : Memorizza i dati grezzi convertiti in digitale dell'ADC di Arduino. Il valore del segnale può variare da 0 a 1024.

- **Int Threshold = 335;** : Corrisponde al valore di soglia, determina quale segnale “contare come battito” e quale ignorare. Si può scegliere un valore per la soglia compreso tra 0 e 1023, ma è consigliato modificarlo con incrementi di 5 o 10. L'abbassamento della soglia rende il sensore più sensibile e viceversa. In questo progetto di tesi sono stati sperimentati valori di soglia differenti fino a che non è stato trovato il migliore.
- Righe 9-14: Funzione di Setup
 - **pinMode(LED13, OUTPUT);** : Configura il pin 13 come uscita per controllare il LED.
 - **Serial.begin(9600);** : Inizializza la comunicazione seriale a 9600 baud per inviare dati al computer.
- Righe 16-33: Funzione Principale (Loop)
 - **Signal = analogRead(PulseSensorPurplePin);** : Legge il valore analogico dal sensore di pulsazioni e lo assegna alla variabile Signal.
 - **Serial.println(Signal);** : Invia il valore del segnale alla porta seriale per visualizzarlo nel Plotter Seriale.
 - **if (Signal > Threshold);** : Controlla se il valore del segnale supera la soglia predefinita. Se sì, accende il LED.
 - **digitalWrite(LED13, HIGH);** : Accende il LED integrato.
 - **else;** : Altrimenti, se il valore del segnale non supera la soglia predefinita,
 - **digitalWrite(LED13, LOW);** : Spegne il LED integrato se il segnale è inferiore alla soglia.
 - **delay(20);** : Introduce un ritardo di 20 millisecondi tra le letture successive per stabilizzare la lettura del segnale.

Questo codice permette di visualizzare i battiti cardiaci facendo lampeggiare il LED integrato di Arduino in sincronia con il battito rilevato dal sensore di pulsazioni. Inoltre, invia i dati del segnale al Plotter Seriale per una visualizzazione grafica.

3.3.1.1 Prove sperimentali per verificare l'efficacia del primo codice per il Pulse Sensor

Utilizzando questo primo codice è stato possibile ottenere una visualizzazione grafica del segnale sul Plotter Seriale. Tuttavia, il sensore utilizzato in questo e nei successivi esperimenti, come verrà approfondito in seguito, ha mostrato difficoltà nel rilevare la frequenza cardiaca dalle dita. Al contrario, è stato possibile ottenere letture accurate posizionando il sensore sul lobo dell'orecchio e sotto la mandibola, dove è percepibile il polso carotideo.

In Figura 35 è rappresentato il tracciato del battito cardiaco rilevato dal lobo dell'orecchio.

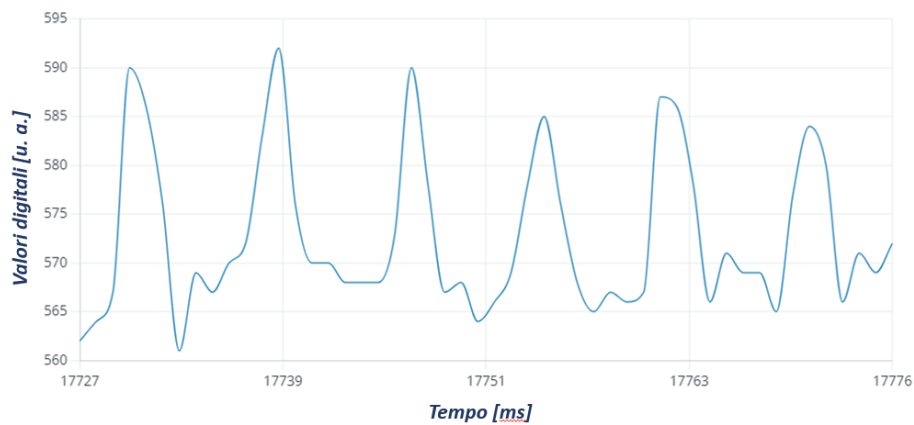


Figura 35: Tracciato del battito cardiaco nel tempo rilevato dal lobo dell'orecchio tramite il Pulse Sensor.

In Figura 36 è rappresentato il tracciato del battito cardiaco rilevato dal polso carotideo.

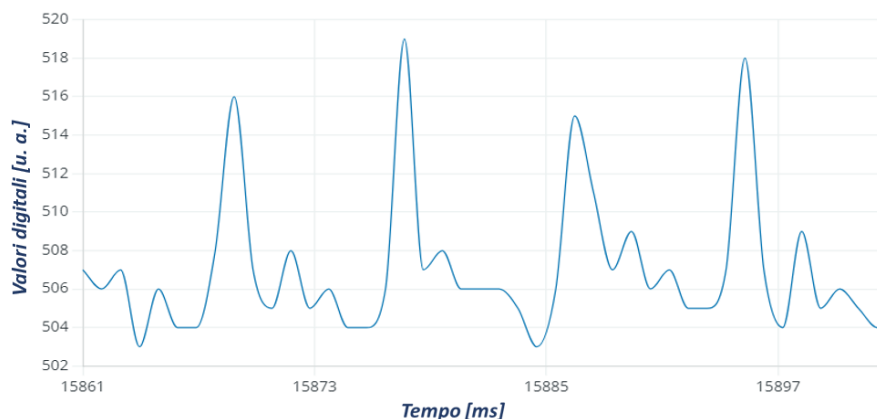


Figura 36: Tracciato del battito cardiaco nel tempo rilevato dal polso carotideo tramite il Pulse Sensor.

durata di vita del sensore, alla sua economicità o alle condizioni dei soggetti. Le prestazioni del sensore sono ottimizzate da un maggiore flusso sanguigno dei soggetti. Nelle giornate fredde o negli utenti con problemi di circolazione (ad esempio con mani fredde), le prestazioni del sensore possono essere influenzate a causa del minor flusso sanguigno.

3.3.3 Terzo Codice: Software “Processing Visualizer”

Questo paragrafo descrive l’uso del “Processing Visualizer”, un software sviluppato per visualizzare i dati raccolti dal Pulse Sensor.

Per utilizzare il “Processing Visualizer”, è necessario seguire alcuni passaggi preliminari:

1. **Installazione della libreria PulseSensor:** Innanzitutto, bisogna installare la libreria “Pulse Sensor Playground” nell’IDE di Arduino.
2. **Modifica del Codice Arduino:** Occorre caricare lo sketch “PulseSensor_BPM” e modificare la variabile “outputType” da SERIAL_PLOTTER a PROCESSING_VISUALIZER. In Figura 38 è mostrato il codice corretto:

```
1  #define USE_ARDUINO_INTERRUPTS true
2  #include <PulseSensorPlayground.h>
3  const int OUTPUT_TYPE = PROCESSING_VISUALIZER;
4  const int PULSE_INPUT = A0;
5  const int PULSE_BLINK = 13;    // Pin 13 is the on-board LED
6  const int PULSE_FADE = 5;
7  const int THRESHOLD = 505;    // Adjust this number to avoid noise when idle
8  PulseSensorPlayground pulseSensor;
9  void setup() {
10     Serial.begin(115200);
11     // Configure the PulseSensor manager.
12     pulseSensor.analogInput(PULSE_INPUT);
13     pulseSensor.blinkOnPulse(PULSE_BLINK);
14     pulseSensor.fadeOnPulse(PULSE_FADE);
15     pulseSensor.setSerial(Serial);
16     pulseSensor.setOutputType(OUTPUT_TYPE);
17     pulseSensor.setThreshold(THRESHOLD);
18     // Now that everything is ready, start reading the PulseSensor signal.
19     if (!pulseSensor.begin()) {
20         for(;;) {
21             // Flash the led to show things didn't work.
22             digitalWrite(PULSE_BLINK, LOW);
23             delay(50);
24             digitalWrite(PULSE_BLINK, HIGH);
25             delay(50);
26         }
27     }
28 }
29 void loop() {
30     delay(20);
31     pulseSensor.outputsSample();
32     if (pulseSensor.sawStartOfBeat()) {
33         pulseSensor.outputBeat();
```

Figura 38: Codice per l’uso di Processing Visualizer.

3. Download del software Processing.

Una volta completata l'installazione, il software Processing può essere avviato. Il "Processing Visualizer" offre un'interfaccia intuitiva per monitorare e analizzare i dati del Pulse Sensor, rendendo più semplice l'interpretazione dei segnali cardiaci.

3.3.3.1 Prove sperimentali per verificare l'efficacia del Processing Visualizer

Dopo aver completato l'implementazione del codice su Arduino UNO e avviato il software "Processing Visualizer", sono stati eseguiti dei test per valutarne l'efficacia e identificare eventuali problematiche.

Il tracciamento dei battiti cardiaci è stato ottenuto posizionando il Pulse Sensor sul pollice, sull'indice, sul lobo dell'orecchio e sotto la mandibola.

Impostando correttamente il parametro di soglia 'Threshold', è stato verificato che il sensore è in grado di misurare accuratamente la frequenza cardiaca dalle dita. Questa accuratezza è stata confermata confrontando visivamente le misurazioni del Pulse Sensor con quelle del pulsossimetro 'Beurer'. Tuttavia, per gli stessi motivi spiegati nel paragrafo 3.3.2.1, è stato possibile ottenere misurazioni corrette dalle dita solo una volta.

In Figura 39 è rappresentato il tracciato del battito cardiaco rilevato dal pollice della mano destra e visualizzato tramite il software "Processing Visualizer".

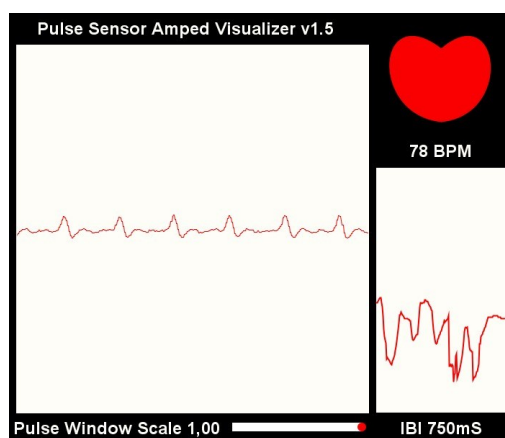


Figura 39: Tracciato del battito cardiaco rilevato dal pollice tramite il Pulse Sensor.

In Figura 40 è rappresentato il tracciato del battito cardiaco rilevato dall'indice della mano destra.

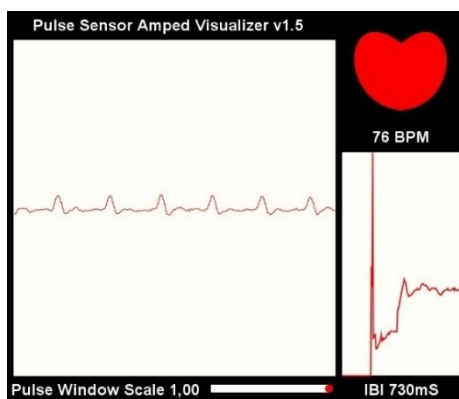


Figura 40: Tracciato del battito cardiaco rilevato dall'indice tramite il Pulse Sensor.

In Figura 41 è rappresentato il tracciato del battito cardiaco rilevato dal lobo dell'orecchio.

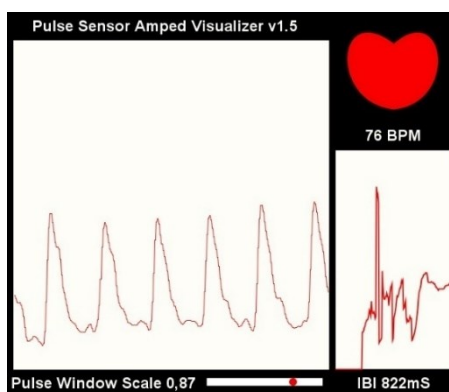


Figura 41: Tracciato del battito cardiaco rilevato dal lobo dell'orecchio tramite il Pulse Sensor.

In Figura 42 è rappresentato il tracciato del battito cardiaco rilevato dal polso carotideo.

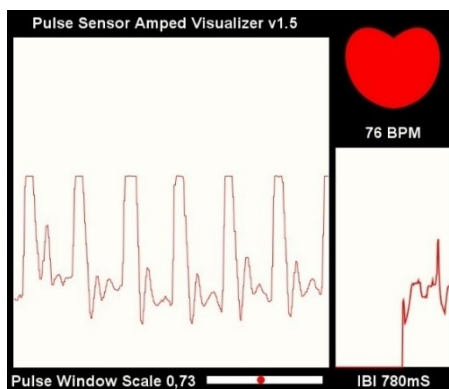


Figura 42: Tracciato del battito cardiaco rilevato dal polso carotideo tramite il Pulse Sensor.

3.4 Sensore MAX30102: Utilizzo della Libreria “Spark Fun Electronics” e Collegamenti del Sensore

In questo paragrafo vengono illustrati i collegamenti e i codici implementati per il funzionamento del sensore MAX30102.

La libreria “SparkFun_MAX3010x” di Arduino contiene una serie di esempi di codice che possono essere utilizzati come base per lo sviluppo di codici personalizzati [14].

Dopo aver collegato il sensore MAX30102, montato su una breadboard, alla scheda Arduino UNO come illustrato al paragrafo 2.5.1, e aver caricato i codici della libreria sul software di Arduino, il sensore ha mostrato malfunzionamenti.

Sono stati testati tre sensori MAX30102 identici, e tutti e tre, una volta collegati al microcontrollore tramite i jumpers, si surriscaldavano e il LED rosso integrato non si accendeva. Pertanto, si può considerare che i sensori MAX30102 testati presentino un malfunzionamento sistematico.

Sono stati effettuati diversi tentativi di collegamento tra il sensore e il microcontrollore; tuttavia, nessuno di essi ha funzionato, nonostante l’uso di vari codici.

Oltre al collegamento illustrato nel paragrafo 2.5.1, sono stati realizzati altri tre collegamenti.

Nel primo collegamento, sono stati utilizzati i codici della libreria di Arduino “SparkFun_MAX3010x” e il codice descritto al paragrafo 3.2.1, relativo al sensore Grove-Finger-Clip. Questi codici sono in grado di leggere anche il sensore MAX30102, ma non riescono a effettuare alcuna misurazione. Il collegamento eseguito per implementare questi codici è illustrato in Figura 43:

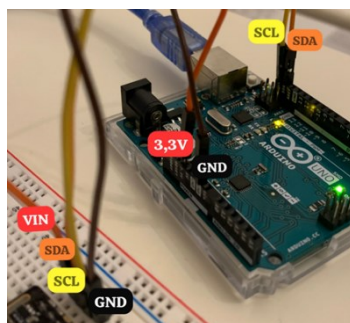


Figura 43: Primo tentativo di collegamento per il MAX30102.

- Il Pin **GND** del sensore è stato collegato, tramite il cavo marrone, al corrispondente Pin di terra della scheda Arduino.
- Il Pin **SDA** del sensore è stato collegato, tramite il cavo arancione, al corrispondente Pin digitale SDA del microcontrollore.
- Il Pin **SCL** del sensore è stato collegato, tramite il cavo giallo, al corrispondente Pin digitale.
- Il Pin **VIN** del sensore è stato collegato, tramite il cavo rosso, al Pin di alimentazione a 3.3 V.

Nel secondo collegamento, è stato utilizzato il codice descritto nel paragrafo 3.2.2, relativo al sensore Grove-Finger-Clip. Questo codice è in grado di leggere anche il sensore MAX30102, ma non riesce ad effettuare le misurazioni. Il collegamento eseguito per implementare questo codici è illustrato in Figura 44:

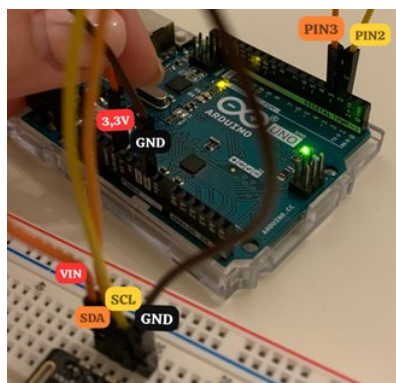


Figura 44: Secondo tentativo di collegamento per il MAX30102.

- Il Pin **GND** del sensore è stato collegato, tramite il cavo marrone, al corrispondente Pin di terra della scheda Arduino.
- Il Pin **SDA** del sensore è stato collegato, tramite il cavo arancione, al Pin3 digitale del microcontrollore.
- Il Pin **SCL** del sensore è stato collegato, tramite il cavo giallo, al Pin2 digitale.
- Il Pin **VIN** del sensore è stato collegato, tramite il cavo rosso, al Pin di alimentazione a 3.3 V del microcontrollore.

Infine, è stato realizzato un terzo collegamento utilizzando la fila dei Pin INT, IRD, RD, GND, al fine di verificare se le due file presenti sul sensore MAX30102 fossero state erroneamente scambiate. Effettuando il collegamento mostrato in Figura 45, il LED rosso si accende, ma il sensore non riesce a leggere i dati.

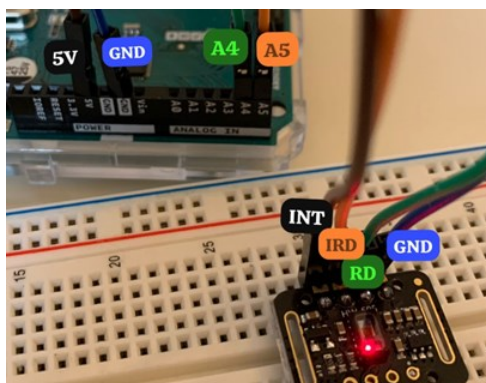


Figura 45: Terzo tentativo di collegamento per il MAX30102.

- Il Pin **GND** del sensore è stato collegato, tramite il cavo blu, al corrispondente Pin di terra della scheda Arduino.
- Il Pin **IRD** del sensore è stato collegato, tramite il cavo arancione, al Pin analogico A4 del microcontrollore.
- Il Pin **RD** del sensore è stato collegato, tramite il cavo verde, al Pin analogico A5.
- Il Pin **VIN** del sensore è stato collegato, tramite il cavo marrone, al Pin di alimentazione a 5 V del microcontrollore.

Capitolo 4: Prove e risultati sperimentali sul Volante Intelligente

In questo capitolo viene descritta l'installazione sul volante del sensore Grove-Finger-Clip, sia con che senza involucro, e le acquisizioni dei dati ottenute durante i test.

4.1 Installazione del sensore Grove – Finger-Clip, con involucro, sul volante

Il sensore Grove – Finger-Clip è stato installato sul volante da gioco nel laboratorio per testarne la funzionalità durante una simulazione di guida.

È stato determinato che la posizione più ergonomica per l'installazione del sensore è quella illustrata in Figura 46.

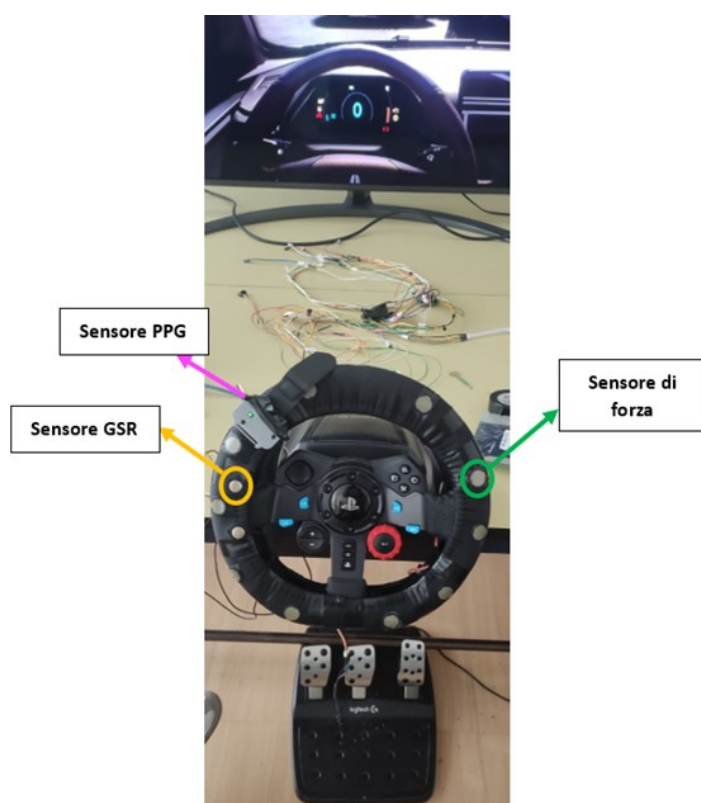


Figura 46: Installazione del sensore Grove-Finger-Clip sul volante.

Per misurare la frequenza cardiaca, l'utente deve appoggiare il pollice della mano sinistra sul sensore.

4.1.1 Risultati sperimentali ottenuti dal Grove – Finger-Clip con involucro

Dopo aver fissato il sensore sul volante (Figura 46), averlo collegato alla scheda Arduino UNO e caricato il codice descritto nel paragrafo 3.2.1, sono stati eseguiti dei test per valutarne l'efficacia e identificare eventuali limitazioni.

Le prove sono state condotte su due soggetti durante una simulazione di guida utilizzando la periferica di gioco (Figura 47).



Figura 47: Simulazione di guida con la periferica di gioco.

In Figura 48 e in Figura 49 sono riportati i grafici che mostrano l'andamento della frequenza cardiaca nel tempo mentre i due soggetti erano alla guida.

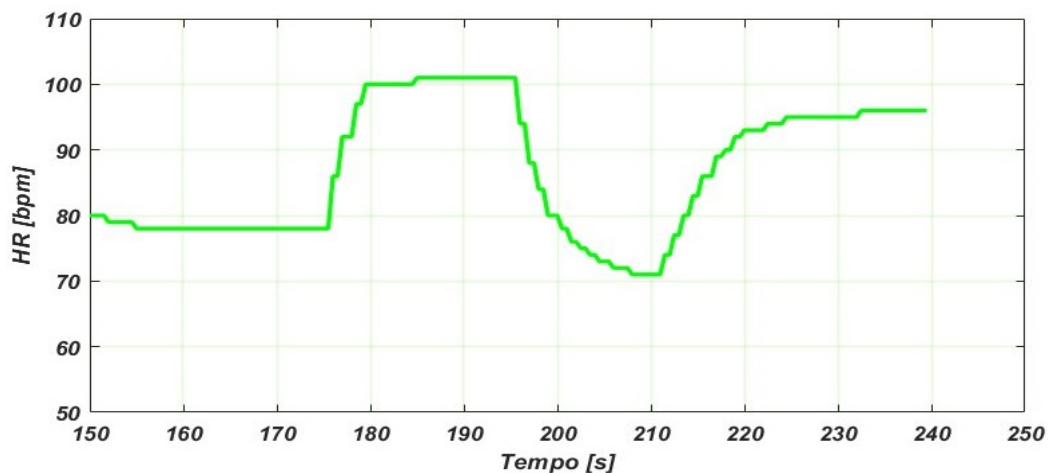


Figura 48: Acquisizione tramite Grove-Finger-Clip su Soggetto 1 alla guida.

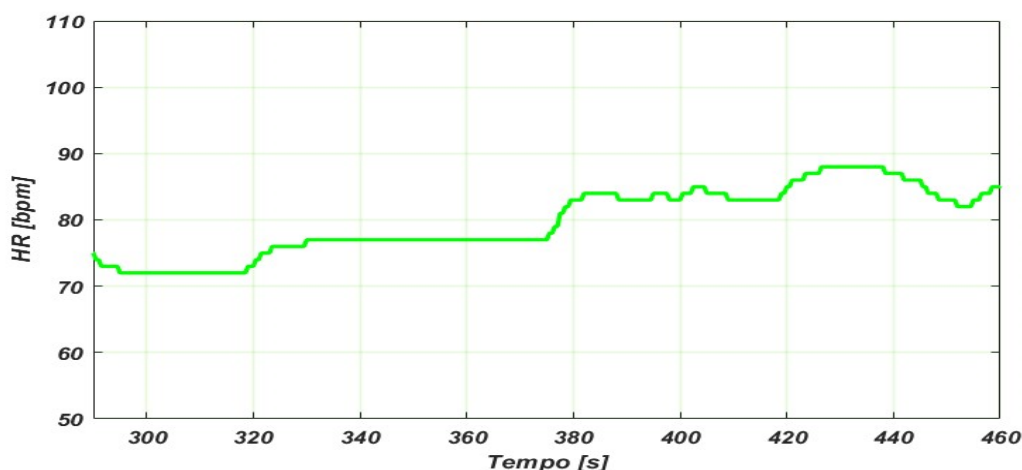


Figura 49: Acquisizione tramite Grove-Finger-Clip su Soggetto 2 alla guida.

In Figura 48, si osserva come la frequenza cardiaca del soggetto 1 sia inizialmente nella norma durante uno stato di quiete. Tuttavia, dopo alcuni secondi, aumenta fino a 100 BPM a causa della tensione dovuta a situazioni di guida stressanti, come l'uscita di strada o un tamponamento.

In Figura 49, si nota un andamento più regolare del battito cardiaco nel tempo, con incrementi dovuti all'emozione e allo stress durante la guida.

In generale, molte delle acquisizioni ottenute dal sensore durante la guida non corrispondevano alla frequenza cardiaca effettiva del conducente. Pertanto, i grafici di Figura 48 e di Figura 49 sono stati rielaborati, omettendo le acquisizioni registrate nei momenti in cui il conducente non manteneva il pollice correttamente sul sensore, dando luogo a dati palesemente corrotti e inutilizzabili. Questa rielaborazione è stata effettuata per permettere una lettura più accurata dei dati.

Come descritto al paragrafo 3.2.3, anche il minimo movimento può compromettere la precisione della misurazione delle pulsazioni cardiache da parte del sensore Grove-Finger-Clip. Per questo motivo si ritiene che il sensore non sia adatto per l'installazione sul volante sensorizzato. La corretta misurazione del battito cardiaco è compromessa per due motivi principali: da un lato, l'involucro realizzato in stampa 3D, che contiene il sensore, è troppo imborgate per essere posizionato correttamente sul volante senza compromettere la comodità del conducente; dall'altro, per ottenere delle acquisizioni accurate, è fondamentale che il sensore sia sempre a stretto contatto con la pelle, cosa

difficile da garantire durante la guida, poiché il conducente tende a spostare le mani sul volante e raramente le mantiene ferme nella stessa posizione.

4.2 Installazione del sensore Grove-Finger-Clip, senza involucro, su un volante provvisorio

Per ovviare al problema dell'eccessivo ingombro del sensore, è stato rimosso l'involucro in stampa 3D, lasciando il sensore privo di rivestimento (Figura 50). In questo modo, sono state ottenute acquisizioni di buona qualità, rendendo il sensore meno invasivo e più adatto all'installazione sul volante.



Figura 50: Sensore Grove-Finger-Clip senza involucro.

L'installazione del sensore privo di involucro è stata eseguita su un volante da gioco per PlayStation, le cui dimensioni risultano essere molto simili a quelle del volante sensorizzato utilizzato in laboratorio. Il sensore è stato montato sul volante in due diverse posizioni al fine di valutare quella più ergonomica.

Inizialmente, il sensore è stato posizionato come illustrato in Figura 51, in modo tale che il conducente potesse poggiare il pollice della mano sinistra sul sensore. Successivamente, il sensore è stato collocato leggermente più in basso (Figura 52), consentendo al guidatore di mantenere il palmo della mano sinistra a contatto con il sensore, permettendo così la misurazione del battito cardiaco attraverso il palmo della mano.



Figura 51: Posizionamento del Grove-Finger-Clip, senza guscio, per poggiare il pollice.



Figura 52: Posizionamento del Grove-Finger-Clip, senza guscio, per poggiare il palmo della mano.

4.2.1 Risultati sperimentali ottenuti dal Grove-Finger-Clip senza involucro

Le acquisizioni con il sensore Grove-Finger-Clip, privo di involucro, sono state effettuate chiedendo ai soggetti testati di muovere il volante imitando i movimenti tipici di chi guida. I soggetti dovevano poggiare il pollice o il palmo della mano sinistra sul sensore installato sul volante, mentre nell'altra mano indossavano il pulsossimetro "Beurer". I dati raccolti dai due dispositivi risultavano coincidenti: le misurazioni del pulsossimetro sono state verificate visivamente, mentre i dati del sensore Grove-Finger-Clip sono stati salvati tramite CoolTerm e successivamente elaborati nei grafici MATLAB riportati nelle figure seguenti.

Le prove sono state eseguite collegando il sensore Grove-Finger-Clip alla scheda Arduino, come descritto al paragrafo 2.3.1.2, e caricando il codice descritto al paragrafo 3.2.2 nel software. I dati acquisiti durante i test, nelle due configurazioni, sono stati elaborati in grafici che mostrano l'andamento del battito cardiaco nel tempo.

La Figura 53 mostra il grafico delle acquisizioni effettuate nella configurazione in cui il soggetto pone il pollice della mano sinistra sul sensore installato sul volante.

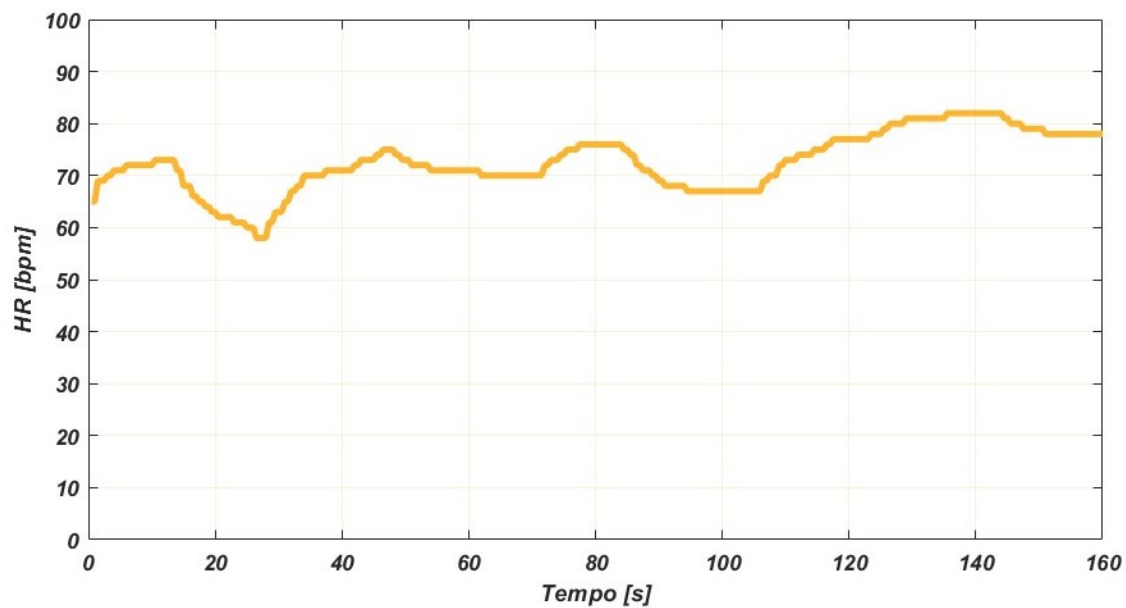


Figura 53: Monitoraggio del battito cardiaco dal pollice tramite il sensore Grove-Finger-Clip privo di involucro.



Figura 54: Test con Sensore PPG a contatto con il pollice.

La Figura 55 rappresenta il grafico delle acquisizioni eseguite nella configurazione in cui il soggetto pone il palmo della mano sinistra sul sensore installato sul volante.

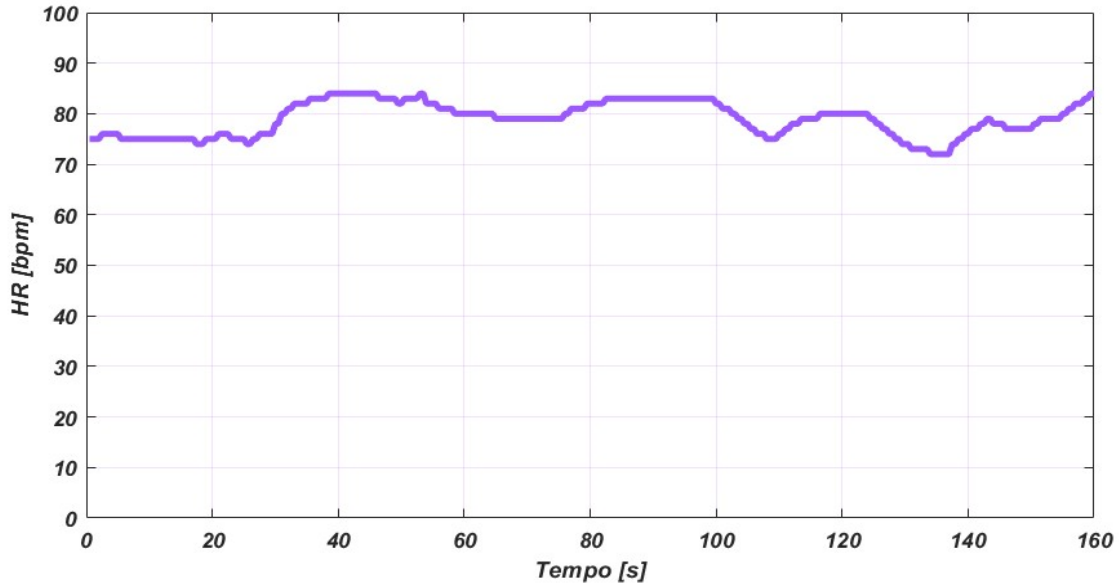


Figura 55: Monitoraggio del battito cardiaco dal palmo della mano tramite il sensore Grove-Finger-Clip privo di involucro.



Figura 56: Test con Sensore PPG a contatto con il palmo della mano.

In seguito, il volante della PlayStation è stato collegato alla relativa console e sono stati acquisiti i dati relativi alla frequenza cardiaca dei soggetti durante l'interazione con il videogioco "F1 23". In Figura 57 è rappresentato il grafico dell'andamento temporale della frequenza cardiaca di un soggetto durante l'interazione con il videogioco, con il palmo della mano costantemente a contatto con il sensore.

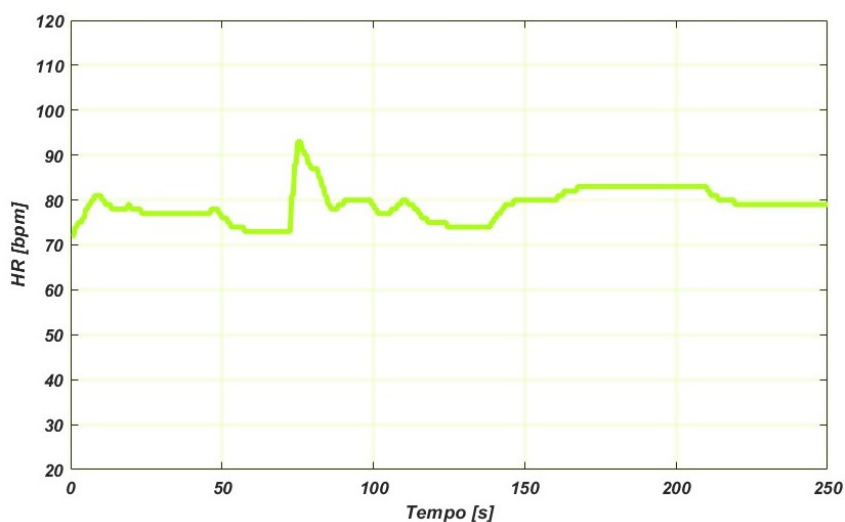


Figura 57: Andamento temporale della frequenza cardiaca durante l'interazione con il videogioco "F1 23" del Soggetto 1.

In Figura 58 è illustrato il grafico dell'andamento temporale della frequenza cardiaca di un soggetto durante l'interazione con il videogioco, senza preoccuparsi di mantenere il palmo della mano costantemente a contatto con il sensore. Di conseguenza, i dati risultano palesemente corrotti e non coincidono con le misurazioni effettuate dal pulsossimetro. Sebbene i grafici del pulsossimetro non siano stati inclusi nella tesi, i dati raccolti sono stati verificati visivamente durante le prove. Questa verifica visiva ha confermato che i dati del pulsossimetro non coincidevano con quelli ottenuti dal sensore, evidenziando così la presenza di dati inutilizzabili.

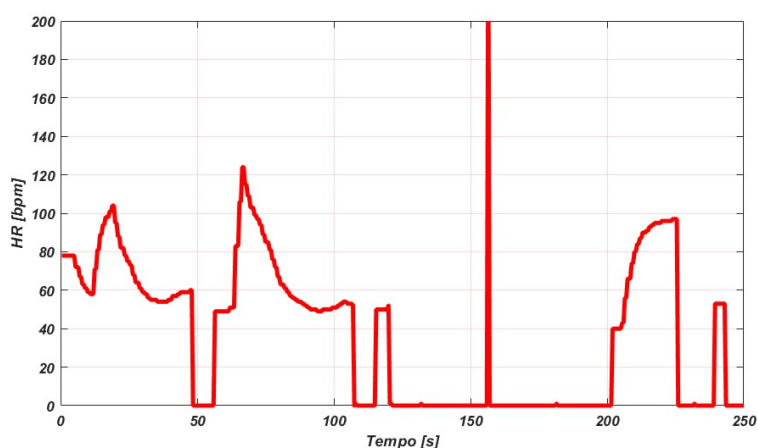


Figura 58: Andamento temporale della frequenza cardiaca durante l'interazione con il videogioco "F1 23" del Soggetto 2.

Capitolo 5: Discussione dei risultati, Conclusioni e Sviluppi Futuri

Nella presente tesi è stata descritta l'implementazione e la verifica sperimentale di tre sensori ottici di frequenza cardiaca: il Grove-Finger-Clip, il Pulse Sensor e il sensore MAX30102. Questi tre sensori sono stati analizzati al fine di stabilire quale fosse il più adatto per l'installazione sul volante sensorizzato.

L'analisi ha incluso una serie di test per valutare la precisione e la reattività di ciascun sensore in condizioni operative simulate. I risultati ottenuti hanno permesso di identificare il sensore più performante, tenendo conto delle specifiche esigenze di integrazione e delle condizioni di utilizzo previste per il volante sensorizzato.

Dai test effettuati, è emerso che i sensori Pulse Sensor e MAX30102 presentano problematiche che ne impediscono un'efficace adozione nella configurazione integrata al volante. Nonostante il loro ingombro minimo li renderebbe particolarmente adatti dal punto di vista della minima invasività e del massimo comfort per il conducente, le difficoltà riscontrate ne limitano l'utilizzo.

In particolare, durante le prove sperimentali, il Pulse Sensor ha mostrato difficoltà nel rilevare il battito cardiaco dalle dita. Questo comportamento è da attribuire a diversi fattori, tra cui la bassa durata di vita del sensore, la sua economicità e le condizioni del flusso sanguigno dei soggetti. In presenza di problemi di circolazione o mani molto fredde, il sensore può avere difficoltà a rilevare correttamente il battito cardiaco.

Per quanto riguarda il sensore MAX30102, i campioni disponibili sono risultati malfunzionanti, poiché tutti e tre i dispositivi MAX30102 testati si surriscaldavano al collegamento con la scheda Arduino Uno, impedendo la corretta lettura della frequenza cardiaca.

Pertanto, il sensore Grove-Finger-Clip, privo di involucro, si è rivelato il più idoneo per l'applicazione proposta, dimostrando di misurare correttamente la frequenza cardiaca dalle dita e dal palmo della mano. Inoltre, ha mostrato un'accuratezza notevole anche durante l'installazione sul volante della PlayStation e nei test effettuati con il videogioco "F1 23". Tuttavia, è importante sottolineare che l'installazione provvisoria sul volante da

gioco non garantiva un comfort ottimale per il guidatore, il quale doveva mantenere il palmo della mano costantemente a contatto con il sensore per evitare acquisizioni di dati corrotti e inutilizzabili.

Un aspetto positivo è che il sensore Grove-Finger-Clip, privo di involucro, se posizionato correttamente, riesce a eseguire misurazioni del battito cardiaco in linea con quelle effettuate dal dispositivo medico “Beurer”, sia dal palmo della mano che dalle dita. Questo risultato evidenzia il potenziale del sensore per applicazioni future, non solo in ambito ludico, ma anche in contesti clinici e di monitoraggio della salute.

In conclusione, il lavoro svolto ha permesso di identificare il sensore più adatto per l’integrazione su un volante intelligente, fornendo una base solida per ulteriori sviluppi futuri e ottimizzazioni. Le problematiche riscontrate con i sensori Pulse Sensor e MAX30102, come la difficoltà nel rilevare il battito cardiaco e il surriscaldamento, indicano la necessità di ulteriori ricerche e miglioramenti tecnologici.

Il sensore Grove-Finger-Clip, invece, rappresenta una soluzione promettente, sebbene siano necessari ulteriori studi per migliorare il comfort e l’usabilità del sistema integrato. Sarà fondamentale installare il sensore Grove-Finger-Clip privo di involucro sulla periferica di gioco del laboratorio, per verificare l’accuratezza delle acquisizioni durante la simulazione di guida e garantire un comfort adeguato al conducente. Inoltre, sarà importante confrontare i segnali rilevati dal sensore PPG con i dati dell’eventuale giroscopio e accelerometro del volante intelligente, per identificare artefatti di movimento nei segnali della frequenza cardiaca causati dalla rotazione del volante e dalle irregolarità della strada. Questo è uno dei motivi per cui le acquisizioni del battito cardiaco nel tempo, presentate nei grafici di *Figura 48*, *Figura 49* e *Figura 57*, sono state rielaborate, omettendo i dati palesemente corrotti e inutilizzabili. Anche disponendo di sensori costosi e altamente performanti, tenendo conto delle specifiche esigenze di integrazione e delle condizioni di utilizzo previste per il volante sensorizzato, sarà difficile ottenere delle acquisizioni completamente pulite. Pertanto, in ogni caso, saranno obbligatori ulteriori controlli e filtraggi.

Inoltre, sarà opportuno esplorare altri sensori per la misurazione della frequenza cardiaca disponibili sul mercato, che potrebbero essere più adatti ai nostri scopi. Questo approccio permetterà di confrontare le prestazioni del Grove-Finger-Clip con quelle di altri

dispositivi, identificando eventualmente soluzioni alternative che offrano una maggiore affidabilità e comfort per l'utente finale.

In sintesi, il lavoro svolto ha gettato le basi per lo sviluppo di un sistema di monitoraggio della frequenza cardiaca integrato in un volante intelligente, evidenziando sia le potenzialità che le limitazioni dei sensori testati. Le future ricerche dovranno concentrarsi sull'ottimizzazione del sensore scelto e sull'esplorazione di nuove tecnologie per migliorare ulteriormente l'esperienza dell'utente.

Bibliografia

- [1] Y. Choi, H. Shin and J. Lee, "Smart steering wheel system for driver's emergency situation using physiological sensors and smart phone," 2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings, Alberobello, Italy, 2014, pp. 281-286, doi: 10.1109/INISTA.2014.6873631.
- [2] Babusiak B.; Hajducik A.; Medvecky S.; Lukac M.; Klarak J. "Design of Smart Steering Wheel for Unobtrusive Health and Drowsiness Monitoring". *Sensors* 2021, 21, 5285. <https://doi.org/10.3390/s21165285>.
- [3] *Introduzione alla piattaforma Arduino: Prototipazione elettronica open-source:* <https://www.giuseppecaccavale.it/corso-arduino/>
- [4] *Guida Introduttiva alla Scheda Arduino UNO R3 e al Microcontrollore ATMEGA328:* <https://www.ne555.it/guida-arduino-scheda-uno/>
- [5] *Grove-Finger-Clip, Sensore di Frequenza Cardiaca a Clip per Dito: Specifiche e Utilizzo:* https://wiki.seeedstudio.com/Grove-Finger-clip_Heart_Rate_Sensor_with_shell/#with-arduino
- [6] *Le Interfacce di Comunicazione della Scheda Arduino:*
[Tutorial – Arduino spiegato facile – Interfacce di comunicazione - Antima](#)
- [7] *Pulse Sensor, Sensore di Battito Cardiaco per Arduino: Guida Completa:* <https://lastminuteengineers.com/pulse-sensor-arduino-tutorial/>
- [8] *Fotoplethismografia: Tecnologia, Analisi del Segnale e Applicazioni Cliniche:* <https://en.wikipedia.org/wiki/Photoplethysmogram>
- [9] *Guida Introduttiva a MATLAB, Funzionalità e Utilizzo:* https://stem.elearning.unipd.it/pluginfile.php/370861/mod_folder/content/0/Introduzione_Matlab.pdf
- [10] *Utilizzo del Sensore di Frequenza Cardiaca Grove - Finger-clip, Guida e Codice di Implementazione:* https://wiki.seeedstudio.com/Grove-Finger-clip_Heart_Rate_Sensor/
- [11] *Utilizzo del Sensore di Frequenza Cardiaca Grove - Finger-clip:* <https://www.pleasedontcode.com/it/blog/comprendere-il-protocollo-i2c-in-parole-semplci>
- [12] *Utilizzo della Comunicazione Seriale con Arduino: Monitoraggio e Debug:* <https://www.maffucci.it/2021/12/28/coolterm-alternative-alla-serial-monitor-di-arduino/>
- [13] *Utilizzo e Funzionamento del Pulsossimetro:* <https://bdi.ie/medical/how-does-a-pulse-oximeter-measure-heart-rate/>

[14] *Utilizzo del Sensore di Oximetria e Frequenza Cardiaca MAX30102 nei Progetti Biometrici:*
https://lastminuteengineers.com/max30102-pulse-oximeter-heart-rate-sensor-arduino-tutorial/?utm_content=cmp-true

Ringraziamenti

Desidero esprimere la mia profonda gratitudine a tutte le persone che hanno contribuito alla realizzazione di questa tesi. In primo luogo, ringrazio sinceramente il mio relatore, il professore Alessandro Terenzi, e la professoressa Susanna Spinsante, per la loro guida, i loro consigli e il loro supporto durante tutto il progetto.

Vi sono grata per avermi permesso di intraprendere questo progetto di tesi, che è stato per me una vera boccata d'aria fresca in un periodo in cui faticavo ad innescare la mia scintilla. Stavo immettendo solo combustibile nel mio motore, dimenticando di espellere i gas di scarico e di aspirare aria nuova. Grazie a voi, ho ritrovato l'equilibrio e la motivazione necessari per completare questo lavoro.