



*Corso di Laurea Magistrale in*  
**Ingegneria Biomedica**

*Relazione di progetto del corso di*  
**Sistemi di misura distribuiti**

**SISTEMA DI MISURA DISTRIBUITO  
PER LA MISURA DELLA  
PULSAZIONE CARDIACA**

***GRUPPO 1***

Cirnelli Simone - Matr. 177084 - s.cirnelli@studenti.unimol.it

Dukic Benedetta - Matr. 175219 - b.dukic@studenti.unimol.it

Ravelli Fabrizio - Matr. 177085 - f.ravelli@studenti.unimol.it

Romano Pierfrancesco - Matr. 177087 - p.romano7@studenti.unimol.it

*Anno Accademico 2023/2024*

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Strumentazione utilizzata</b>	<b>2</b>
2.1	Setup di misura . . . . .	3
<b>3</b>	<b>Client</b>	<b>4</b>
3.1	CLIENT: Manuale utente . . . . .	4
3.2	CLIENT: Manuale programmatore . . . . .	7
<b>4</b>	<b>Server</b>	<b>19</b>
4.1	SERVER: Manuale utente . . . . .	20
4.2	SERVER: Manuale programmatore . . . . .	21
<b>5</b>	<b>Esempio di funzionamento e bug riscontrati</b>	<b>28</b>
5.1	Esempio di funzionamento . . . . .	28
5.2	Bug riscontrati . . . . .	31

# 1 Introduzione



Il progetto seguente propone l'implementazione di un sistema di misure distribuito, finalizzato al monitoraggio della frequenza cardiaca, mediante l'utilizzo di un sensore pulsossimetro.

Nel contesto di tale sistema, il sensore è soggetto a un controllo preciso di tensione e corrente, gestito da un alimentatore regolato a distanza tramite interfaccia IEEE-488. Il sensore stesso, è poi connesso ad una scheda di acquisizione dati, consentendone così l'analisi e l'estrinsecazione delle curve di interesse relative alla pulsazione cardiaca.

Attraverso l'implementazione di un sistema di comunicazione server-client connessi tramite protocollo TCP/IP, il presente progetto si articola nella configurazione di due interfacce utenti distintive:

- L'interfaccia lato server è progettata esclusivamente per la visualizzazione del segnale acquisito, finalizzata a verificarne la veridicità ed efficienza in fase di acquisizione. Tale visualizzazione comprende informazioni quali tempo di osservazione e la frequenza di campionamento.
- Dall'altro lato l'interfaccia client si presenta graficamente basata su un design user-friendly dove gli utenti hanno la possibilità di impostare una password per accedere e avviare la procedura di misurazione. La misura stessa è inizializzata all'interno dell'interfaccia client, dove è possibile monitorarne lo stadio di esecuzione temporale. In quest'ultima, vengono visualizzati grafici nel dominio del tempo e delle frequenze e il valore misurato della pulsazione corrente. Inoltre, è implementato un sistema di avviso che segnala il superamento dell'intervallo fisiologico della pulsazione cardiaca.

Risulta necessaria e doverosa, quindi, una breve contestualizzazione del problema dal punto di vista clinico. Il pulsossimetro è, infatti, uno strumento ampiamente impiegato per monitorare la saturazione di ossigeno nel sangue e, di conseguenza, il battito cardiaco. Esso opera mediante la misurazione della luce assorbita dal sangue, fornendo così una valutazione indiretta della frequenza cardiaca. Il segnale acquisito è analizzato in relazione al normale intervallo in cui il ritmo cardiaco fisiologico si colloca.

Nel caso in cui questo valore si discosti da tale limite, si differenzia tra bradicardia (frequenza cardiaca anormalmente bassa) e tachicardia (frequenza cardiaca anormalmente elevata). Fornendo così informazioni fondamentali per la salute del sistema cardiovascolare.

Nel contesto del seguente progetto, l'infrastruttura software adottata comprende LabVIEW 2023 e NI-MAX, integrati con il pacchetto NI-DAQmx e la versione 3.9 di Python.

## 2 Strumentazione utilizzata

In questa sezione vengono riportate le caratteristiche principali e una breve descrizione della strumentazione utilizzata (Tabella 1).

STRUMENTAZIONE	
NOME	Q.TA'
HP Agilent Keysight E3631A	1
Pulsiossimetro riflessivo PulseSensor	1
NI DAQMX USB-6001	1
Interfaccia USB-GPIB	1
Clip a coccodrillo	2
Cavo GPIB/IEEE-488, Keysight	1

Tabella 1: Strumentazione utilizzata

In figura 1 sono riportati gli strumenti utilizzati.



Figura 1: Strumentazione utilizzata.

L'alimentatore programmabile stabilizzato **HP Agilent Keysight E3631A** presenta tre uscite differenti (P6V, P25V e N25V). Lo strumento presenta interfacce GPIB e RS232 ed in grado di erogare una corrente massima di 5 A sul canale P6V e 1 A nei canali P/N25V.

Il **pulsiossimetro riflessivo PulseSensor** è un sensore per la misurazione del battito cardiaco progettato per essere integrato con Arduino ed altri microcontrollori.

Può essere alimentato con tensioni comprese tra  $3 \div 5 < V$  e correnti di  $3 \div 4 \text{ mA}$ . Presenta tre cavi di uscita (GND, SIGNAL, VDD). In figura 2 sono mostrate le dimensioni del sensore.

La scheda di acquisizione dati **NI DAQMX USB-6001** permette di acquisire segnali analogici e convertirli in segnali digitali. Presenta un fondo scala non variabile di  $\pm 10 \text{ V}$ , una frequenza di

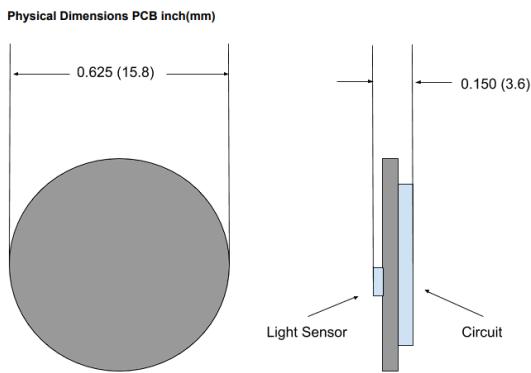


Figura 2: Dimensioni del sensore PulseSensor

campionamento massima pari a 20 kSPS<sup>1</sup> per canale e un convertitore AD a 14 bit. Permette una configurazione plug-and-play tramite connessione USB.

L'**interfaccia USB-GPIB** e il **cavo GPIB/IEEE-488** sono fondamentali per la connessione del nodo di misura al calcolatore deputato al ruolo di server.

## 2.1 Setup di misura

In figura 3 viene riportato schematicamente il setup adoperato. Il cavo GPIB/IEEE-488 è stato utilizzato unicamente per rendere possibile la connessione dell'interfaccia USB-GPIB in quanto lo spazio presente nel pannello posteriore dell'HP Agilent Keysight E3631A non era sufficiente.

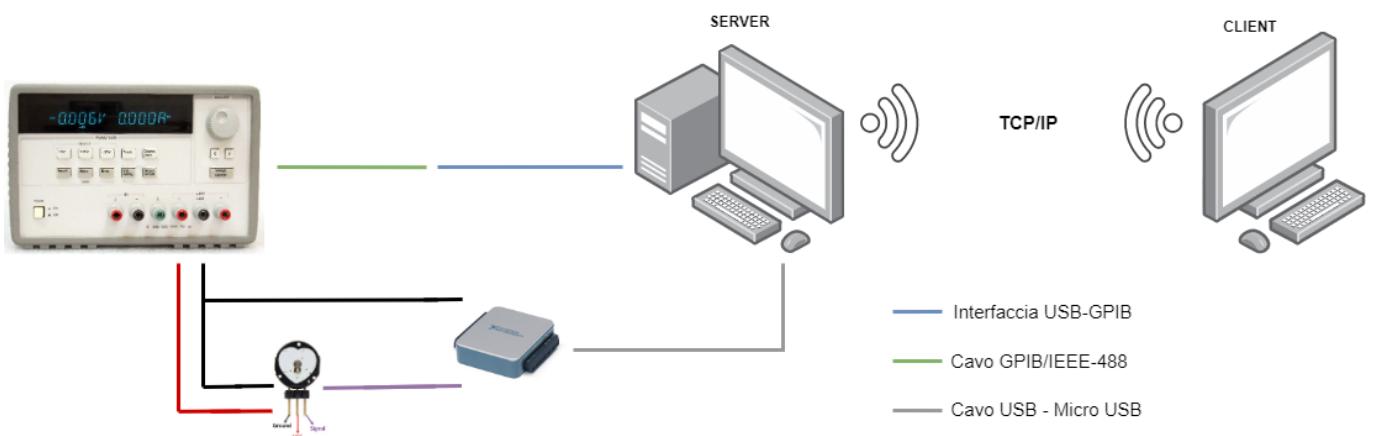


Figura 3: Setup operativo

<sup>1</sup>kSPS: k-Sample Per Secondo

## 3 Client

### 3.1 CLIENT: Manuale utente

In questa sezione l'intento è quello di offrire una panoramica dell'interfaccia con la quale è chiamato ad interagire un generico utente, fornendogli una guida completa sui vari elementi visivi e comandi presenti sullo schermo. L'interfaccia si compone di diverse schede, ciascuna delle quali svolge un ruolo specifico nel processo interattivo. Inizialmente, nella scheda **LOG IN** l'utente autentica la propria identità attraverso l'inserimento di una password dedicata. Soltanto in caso di correttezza della password, viene abilitata la **HOME PAGE** dell'interfaccia, punto centrale per la configurazione dei parametri di misura. Qui è possibile settare le impostazioni desiderate e visualizzare i grafici relativi al segnale acquisito ed elaborato, inclusa la misurazione della pulsazione cardiaca. Inoltre, la scheda denominata **HELP** fornisce dettagliate informazioni su ciascuna sezione dell'interfaccia. Qui, gli utenti possono trovare utili indicazioni, suggerimenti e istruzioni passo-passo relativi alle operazioni necessarie su ogni scheda per ottenere un corretto interfacciamento e un'effettiva visione dei dati misurati. Di seguito ciascuna scheda viene approfondita nel dettaglio.

#### LOG IN

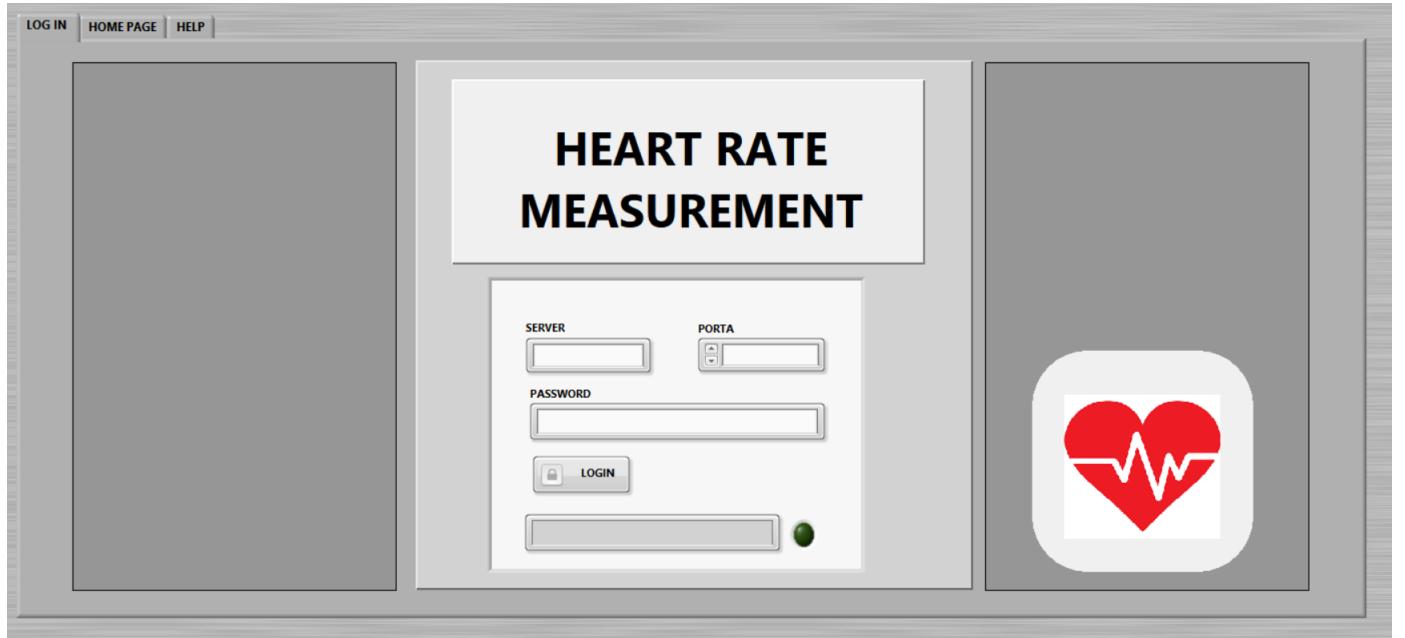


Figura 4: Interfaccia della scheda LOG IN

La scheda di interfaccia iniziale (Figura 4), nota come "LOG IN", costituisce il punto iniziale dell'utilizzo dell'apparecchiatura, introducendo gli utenti alla fase iniziale dell'operatività. Essa infatti prelude alla procedura di misurazione del battito cardiaco a patto che però l'utente inserisca le giuste credenziali. In particolare, oltre alla password, l'utente è chiamato a fornire l'indirizzo IP del server e a specificare la porta di accesso.

Di seguito vengono elencati i controlli che devono essere necessariamente impostati dall'utente prima di cliccare sul pulsante di LOGIN:

- SERVER: tale controllo permette di inserire l'indirizzo IP del server di destinazione (ad esempio 192.168.123.132) o di inserire la dicitura *LOCALHOST* qualora il PC funga sia da server che da client.
- PORT: qui va inserito il numero di porta (può variare da 0 a 65535) per consentire che avvenga la comunicazione tra server e client.
- PASSWORD: serve per digitare l'apposita password che permette di accedere alla scheda successiva di acquisizione del battito. In fase di scrittura essa viene opportunamente nascondata tramite “masked password display” (caratteri rappresentati da asterischi) garantendo una maggiore privacy per l'utente.

Infine, due indicatori forniscono dettagli significativi inerenti allo stato di validità della password e quindi sulla riuscita della connessione client-server. L'illuminazione del LED verde segnala una corretta verifica della password, autorizzando l'utente. In alternativa, un LED spento indica un fallimento nella verifica, accompagnato da un messaggio di errore come “*connection refused* ” anziché “*connection set*”.

## HOME PAGE

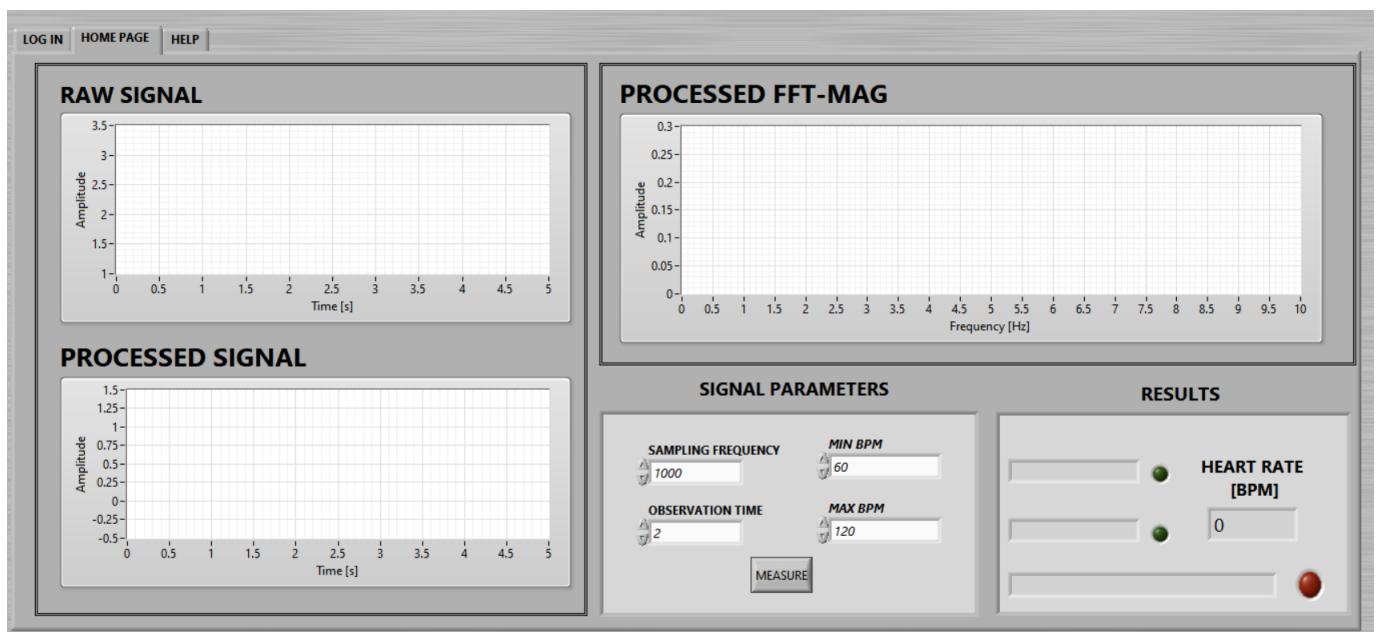


Figura 5: Interfaccia della scheda HOME PAGE

Nella scheda successiva (Figura 5), denominata “HOME PAGE” è possibile definire con precisione i parametri di misura per l’acquisizione del segnale e visualizzare in modo dettagliato l’andamento del segnale acquisito e post-elaborato, nonché il segnale processato nel dominio della frequenza, insieme al valore di pulsazione cardiaca in battiti per minuto (BPM).

La sezione denominata “SIGNAL PARAMETERS” consente all’utente di configurare la frequenza di campionamento e il tempo di osservazione per l’esecuzione della misura. Inoltre, è possibile definire i valori limite di soglia, sia minimo (MIN BPM) che massimo (MAX BPM). Il pulsante “MEASURE” permette di caricare i parametri impostati e avviare il processo di acquisizione.

La sezione dei “RESULTS” presenta vari indicatori, con particolare attenzione al valore effettivo della misura, espresso in BPM come “HEART RATE”. Inoltre, sono presenti messaggi di stato che segnalano l’avvio e la terminazione della misura, o eventuali messaggi di errore qualora si verifichi un problema:

- *Starting measure*: indicatore che esprime una segnalazione dal server di avvio della misura;
- *Measure terminated*: indicatore che esprime una segnalazione dal server di terminazione della misura;
- *ERROR*: indicatore che serve ad informare l’utente del riscontro di un potenziale errore durante l’acquisizione del segnale prima di essere riuscito ad effettuare la misurazione.

Inoltre, un ulteriore indicatore affiancato da un LED ROSSO funge da avviso nel caso in cui il valore di pulsazione rilevato si collochi al di fuori dell’intervallo impostato sull’interfaccia grafica dall’utente. In questo caso sono 3 i possibili messaggi che possono comparire:

- *BRACHICARDICO*: il valore di pulsazione cardiaca misurato risulta inferiore al valore limite (MIN BPM) impostato dall’utente (60 BPM di default) ;
- *NORMAL*: il valore di pulsazione cardiaca misurato risulta interno all’intervallo di valori limite (MIN BPM e MAX BPM) impostato dall’utente;
- *TACHICARDICO*: il valore di pulsazione cardiaca misurato risulta maggiore del valore limite (MAX BPM) impostato dall’utente (120 BPM di default) ;

Infine, la scheda presenta tre grafici informativi: “RAW SIGNAL”, “PROCESSED SIGNAL”, e “PROCESSED FFT-MAG”. Questi consentono una chiara visualizzazione dell’andamento del segnale nel dominio del tempo e della frequenza, offrendo una rappresentazione visiva dettagliata e immediata dell’analisi del segnale cardiaco. Il segnale grezzo acquisito dal sensore (RAW SIGNAL) viene opportunamente elaborato ottenendo un segnale processato nel tempo (PROCESSED SIGNAL) e poi di esso viene effettuata un’analisi nel dominio della frequenza (PROCESSED FFT-MAG) per ricavare direttamente il valore di pulsazione cardiaca.

## HELP

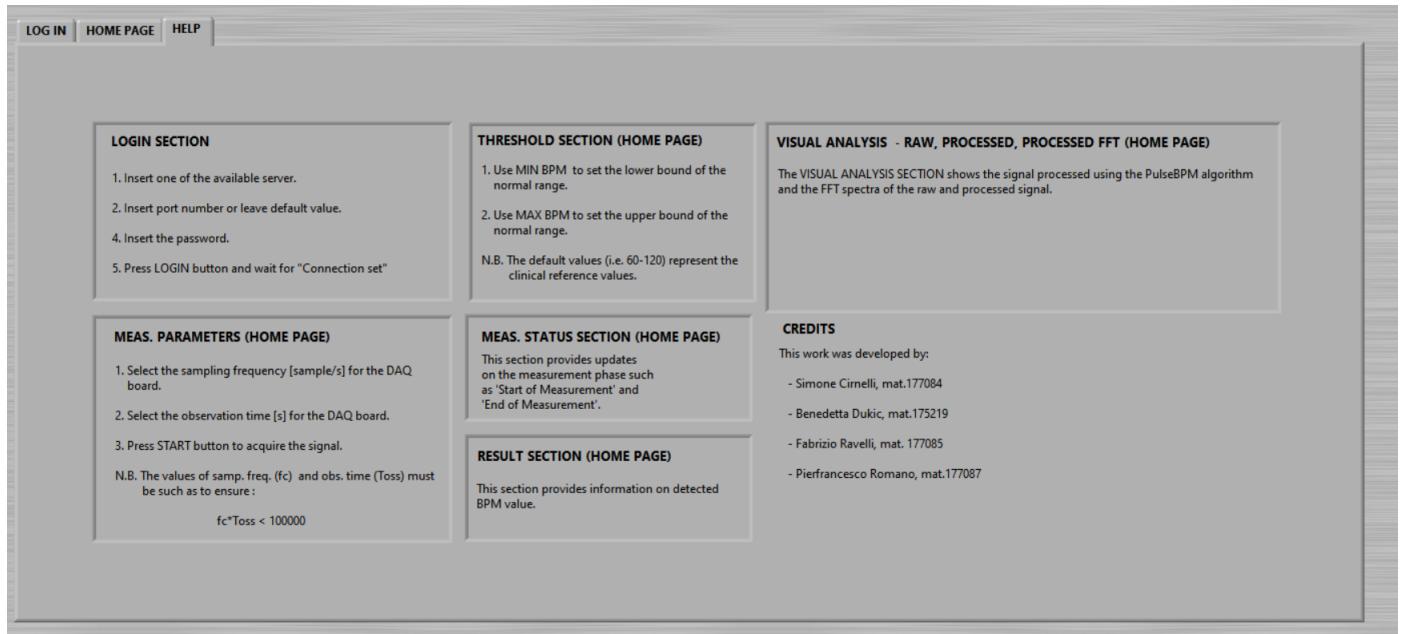


Figura 6: Interfaccia della scheda HELP

La scheda HELP (Figura 6) è progettata per guidare l'utente attraverso il processo di misurazione, offrendo informazioni dettagliate e suggerimenti suddivisi in piccoli riquadri, ciascuno dedicato a una specifica sezione. Questi riquadri offrono una guida chiara passo dopo passo, fornendo all'utente le indicazioni necessarie per eseguire correttamente la misurazione e interpretare i risultati ottenuti dall'apparecchiatura.

### 3.2 CLIENT: Manuale programmatore

Nel contesto progettuale consistente nella realizzazione di un misuratore di pulsazione cardiaca, il Block Diagram di LabVIEW rappresenta un elemento cruciale dell'implementazione del lato Client. Attraverso una serie di nodi, terminali e funzioni ben definiti, viene garantita un'interazione client-server fluida e sicura, contribuendo a creare un sistema di misurazione avanzato e affidabile.

Nella descrizione dettagliata che verrà di seguito proposta, vengono esplorate in modo approfondito le varie sezioni del Block Diagram, analizzando le varie fasi del processo di misurazione della pulsazione cardiaca. Si parte dall'accesso sicuro attraverso la crittografia della password e dalla verifica delle credenziali, per poi proseguire con le fasi della misurazione stessa, dall'invio dei parametri di segnale fino alla visualizzazione dei risultati elaborati. L'obiettivo è quello di offrire un'analisi chiara e comprensibile di come il Block Diagram di LabVIEW supporti ogni aspetto applicativo, mostrando come la potenza e la flessibilità di LabVIEW siano state sfruttate per realizzare un sistema robusto ed avanzato.

Il Block Diagram della nostra applicazione di misurazione della pulsazione cardiaca può essere chiaramente suddiviso in due zone distinte, ciascuna dedicata a una specifica fase del processo. La prima zona è dedicata alla fase di autenticazione e accesso, rappresentata dalla scheda "LOG IN"(Figura 7), mentre la seconda zona si focalizza sulla "HOME PAGE", dove avvengono la configurazione e la visualizzazione dei risultati della misurazione (Figura 8).

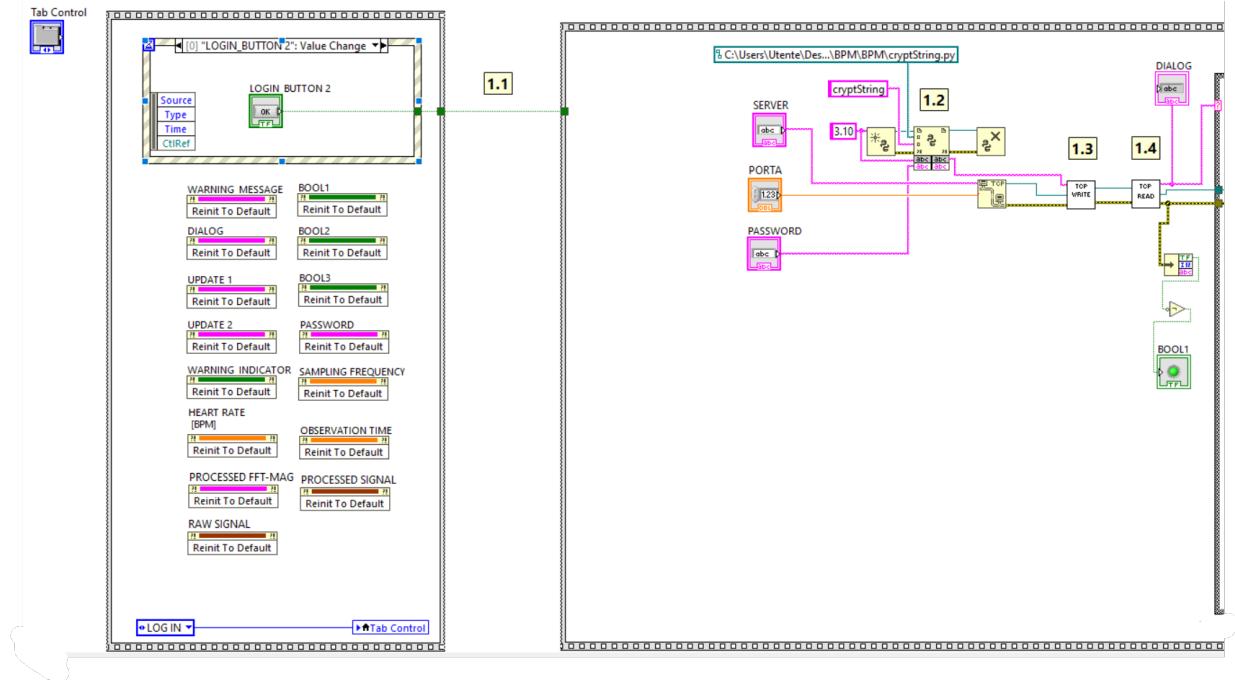


Figura 7: Block Diagram relativo alla scheda di 'LOG IN'

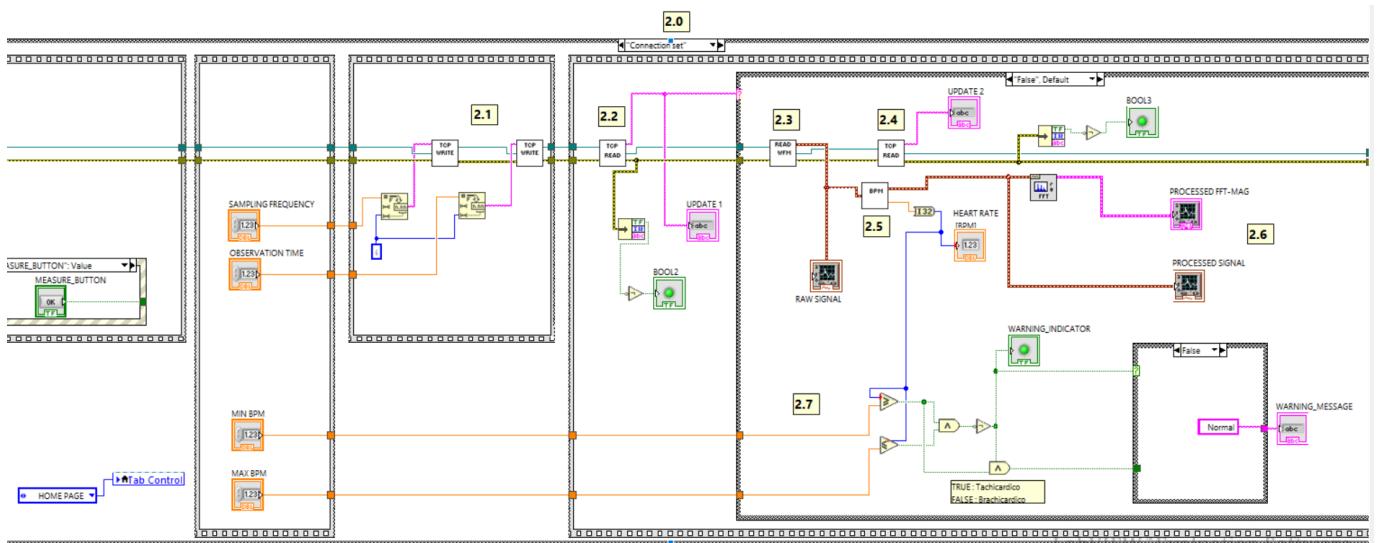


Figura 8: Block Diagram relativo alla scheda di 'HOME PAGE'

Le fasi distinte del flusso operativo sono state enumerate per una più facile comprensione del processo di misurazione, fornendo una mappa chiara e logica della attività svolte dal sistema in esame. Di seguito vengono mostrati, così come appaiono nel Block Diagram, i passaggi logici da seguire (Figure 9 - 10).

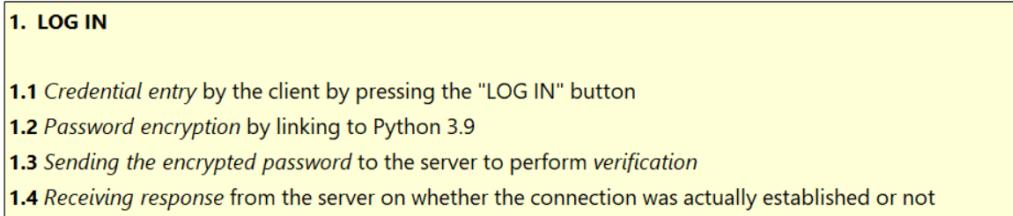


Figura 9: Fasi del flusso operativo relative alla scheda di "LOG IN"

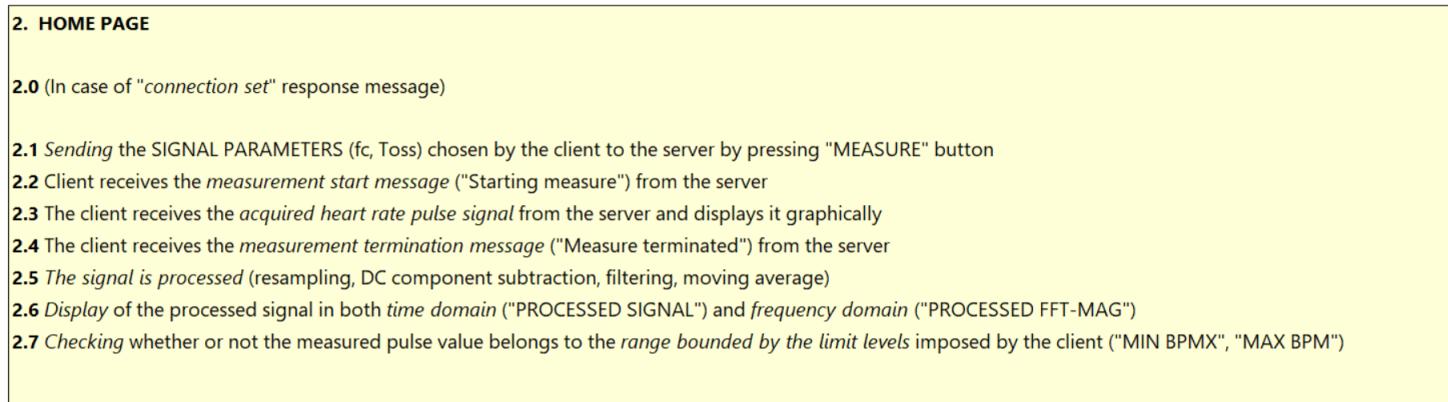


Figura 10: Fasi del flusso operativo relative alla scheda di "HOME PAGE"

All'interno del Block Diagram, per garantire un design modulare e una gestione efficiente delle operazioni ripetitive, viene fatto ampio uso dei **SubVI**. Un subVI è una porzione di codice LabVIEW incorporato all'interno di un diagramma a blocchi principale. Funge da sottoprogramma, consentendo la modularità e la riutilizzabilità del codice. Esso esegue una specifica funzione o operazione ed è richiamato dal diagramma principale per semplificare la progettazione. È qui opportuno andarli a descrivere singolarmente prima di procedere alla spiegazione delle fasi di processo dato che ricorrono più volte all'interno del Block Diagram e in modo da non risultare ripetitivi nella descrizione. Di seguito un elenco dei SubVI presenti dal lato client (alcuni dei quali ospitati anche nel lato server):

- *TCP Write;*
- *TCP Read;*
- *Read WFM;*
- *BPM*

Il *SubVI TCP Write* è responsabile dell'invio di dati al server attraverso il protocollo TCP, mentre il *SubVI TCP Read* è dedicato alla ricezione di dati dal server attraverso il protocollo TCP.

Il primo (Figura 11) prende in input il dato in formato stringa da inviare (*data*), il parametro che identifica in modo specifico la connessione TCP attraverso la quale verranno inviati i dati (*connection ID*) e un cluster in input (*error in*) che può accettare errori di informazione provenienti da VI precedentemente chiamati. In output restituisce lo stesso parametro di connessione (*connection ID out*) in modo da restituire l'identificatore univoco della connessione utilizzato per l'operazione di scrittura e un cluster di errore (*error out*) che monitora l'eventuale comparsa di errori.

Il *subVI TCP Read* (Figura 12) presenta una sintassi speculare al corrispettivo *subVI TCP Write*, avendo gli output di quest'ultimo come input e viceversa. In entrambi compare un blocco di *type cast* per la conversione in stringa e si utilizzano due blocchi *TCP Write* nel *subVI TCP Write* (così come due blocchi *TCP Read* nel *subVI TCP Read*) dato che uno serve per inviare (ricevere) informazioni sulla lunghezza dei dati mentre l'altro per i dati stessi. Anche per il *subVI TCP Read* il *timeout* è impostato al valore di default di -1 (wait forever) che permette di avere il server sempre in attesa. Una costante numerica intera di valore 4 è collegata all'ingresso *bytes to read* del primo *TCP Read* (la funzione *String length* nel *subVI TCP Write* codifica l'uscita su un integer da 32 bit, cioè 4 byte). Con il mode *Buffered* si aspetta fino a quando non saranno disponibili tutti i byte richiesti oppure finché non avviene il time out. Se non venissero inviati esattamente i caratteri in numero atteso allora come dato in output tale modalità può restituire zero bytes.

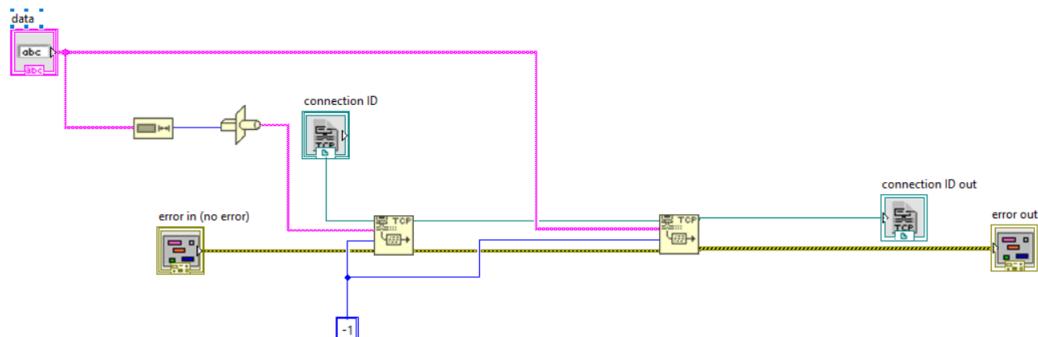


Figura 11: Block Diagram del subVI “TCP Write”

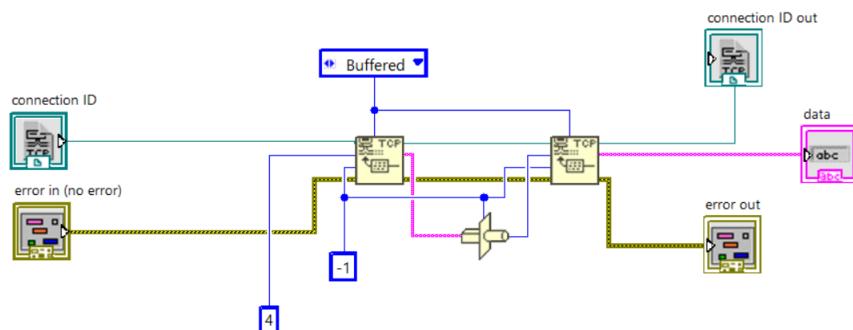


Figura 12: Block Diagram del subVI “TCP Read”

Il SubVI *Read WFM* (Figure 13) è progettato per effettuare la ricezione del segnale di pulsazione cardiaca acquisito dal server ed inviato poi al client. Composto da tre SubVI *TCP Read* distinti, il SubVI acquisisce il tempo di campionamento (*dt*), il numero di campioni (*# samples*) e i campioni stessi (*samples*). Come input tale subVI presenta *Connection ID in* ed *Error in*, mentre come output, oltre a *Connection ID out* ed *Error out*, vi è il segnale di pulsazione cardiaca in formato *Waveform*.

La peculiarità di questo SubVI risiede nell'utilizzo della funzione LabVIEW denominata “*Unflatten to String*”, mirata a ottimizzare il trasferimento efficiente dei dati attraverso la connessione TCP. È importante sottolineare, infatti, come per superare le limitazioni temporali di trasferimento dei dati del protocollo TCP, questo blocchetto risulti alquanto vantaggioso. Tale funzione “*Unflatten from String*” converte in modo efficiente il dato precedentemente “appiattito” (*flattened*) in una rappresentazione numerica interpretabile. Questo approccio consente di gestire in modo ottimale la trasmissione dei dati, riducendo il carico sulla connessione TCP e accelerando il processo di visualizzazione del grafico nel client.

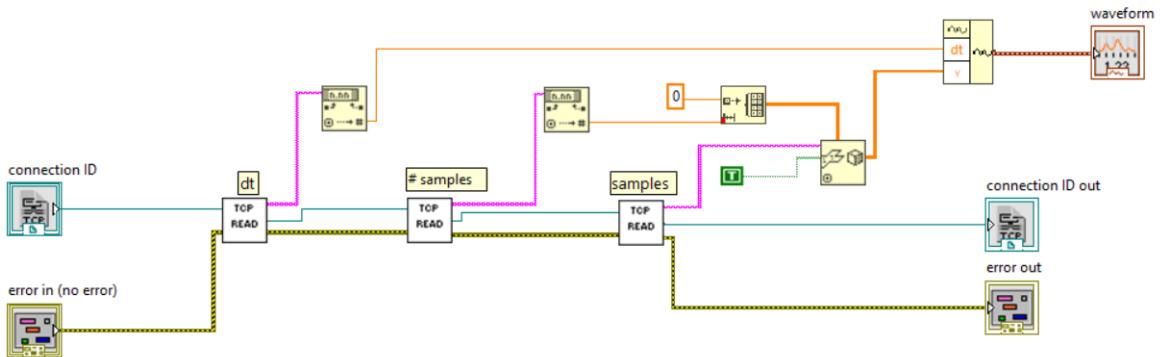


Figura 13: Block Diagram del subVI “Read WFM”

Il SubVI “BPM” rappresenta una tappa cruciale nell’elaborazione del segnale grezzo acquisito dalla scheda di acquisizione, preparandolo per il calcolo accurato della pulsazione cardiaca. Questa fase di condizionamento e processing del segnale è suddivisa in diverse operazioni chiave, mirate a migliorare la qualità e la precisione del segnale di heart rate.

In particolare, in input esso riceve il segnale grezzo acquisito dalla scheda di acquisizione, contenente sia il segnale utile di pulsazione cardiaca che componenti indesiderate come la componente DC e possibili rumori (*waveform in*). In output tale subVI restituisce non solo il segnale elaborato (*waveform out*) ma anche già direttamente il valore di pulsazione cardiaca in BPM (*Heart rate*). Infatti, la funzione LabVIEW “Extract Single Tone Information” permette di estrarre dal segnale in formato *waveform* il tono a più alta intensità. Poi il valore in frequenza ad esso associato viene moltiplicato per 60, ottenendo la pulsazione cardiaca in BPM.

La fase di elaborazione, così come descritto in Figura 14, prevede varie fasi che vengono di seguito descritte:

- *Resampling a 25 HZ*: Utilizzando la funzione LabVIEW “Resample Waveforms”, il segnale viene campionato nuovamente a una frequenza di 25 Hz. Questa operazione riduce il numero di punti nel segnale, ma conserva comunque l’informazione rilevante per il calcolo della pulsazione cardiaca, considerando che la frequenza massima del segnale cardiaco è di solito inferiore a 3 Hz;
- *Sottrazione della componente DC*: La componente DC, non corrispondente al segnale utile ma rappresentante un offset, viene sottratta al segnale acquisito. La sua eventuale presenza farebbe comparire nel grafico FFT del segnale un picco elevato nell’intorno dei 0 HZ non corrispondente al segnale utile di pulsazione cardiaca. Questo passo, quindi, è fondamentale per eliminare componenti non desiderate e preparare il segnale per le successive elaborazioni;
- *Filtraggio Passa-Basso a 3 Hz*: Un filtro passa-basso del primo ordine (*order* impostato ad 1) viene applicato al segnale con la funzione “Butterworth Filter” di LabVIEW. La frequenza di taglio del filtro è fissata a 3 Hz (corrispondente in termini di frequenza a 180 BPM), poiché frequenze superiori non sono rilevanti per una valutazione clinica appropriata dell’heart rate. Questo passo riduce ulteriormente eventuali rumori ad alta frequenza nel segnale;
- *Media mobile con Metodo Henderson*: Infine, il segnale passa attraverso una media mobile utilizzando il blocco “TSA Moving Average” di LabVIEW con il metodo Henderson. Questo passo contribuisce a rendere il segnale più “smooth”, attenuando variazioni rapide e rendendo più evidenti i cambiamenti a lungo termine nel segnale.

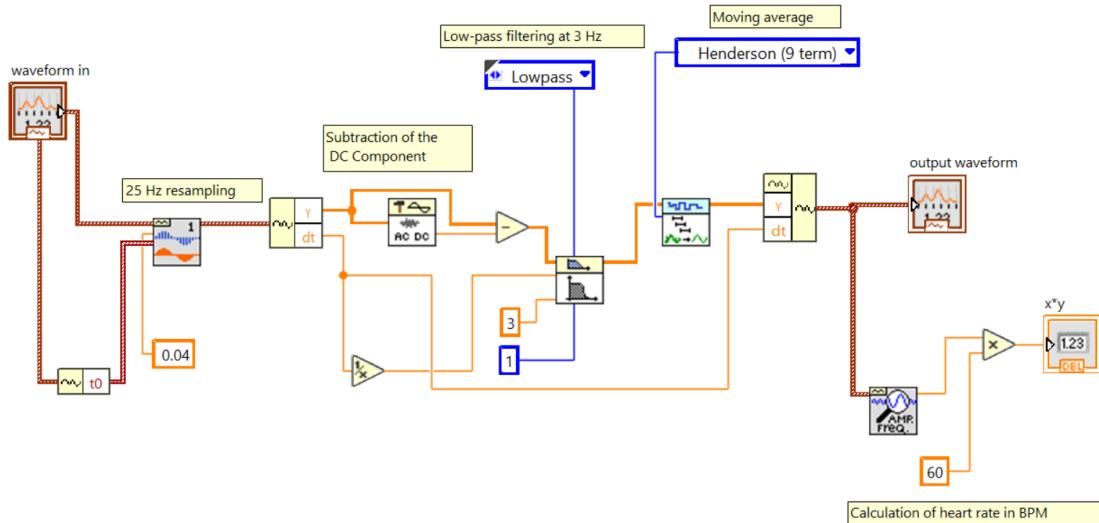


Figura 14: Block Diagram del subVI “BPM”

Dopo aver descritto nel dettaglio le caratteristiche dei subVI, è possibile procedere discutendo del processo di misurazione. Guardando in generale il Block Diagram del lato client va notato che le due schede, LOG IN e HOME PAGE, sono integrate in due frame distinti all'interno della funzione Sequence di LabVIEW. La funzione *Sequence* consente di controllare secondo un flusso ordinato e ben definito l'esecuzione degli elementi presenti nei vari frame che si susseguono. Ciò significa che l'esecuzione segue una sequenza specifica: prima viene elaborato ciò che è contenuto nella scheda di LOG IN e successivamente la parte inserita nella scheda di HOME PAGE.

Nel frame relativo alla scheda di LOG IN, è presente una struttura denominata “Event” in LabVIEW (vedi Figura 15). Tale struttura è progettata per gestire gli eventi, e può entrare in uno stato di timeout in attesa della notifica di un evento specifico. All'interno di questa struttura è stato inserito un pulsante di LOG IN, il cui stato passa da “false” a “true” al momento della pressione. Questo pulsante è cruciale poiché consente di avanzare al frame successivo. In assenza di questa azione, l'esecuzione rimarrebbe in sospeso.

Nello stesso frame, sono stati inclusi blocchetti “Invoke method: Renitalize to default” relativi a diversi indicatori e controlli. Questi blocchi hanno la funzione di riportare tali elementi al valore di default. Questa operazione è particolarmente utile per ripulire eventuali residui o caricamenti derivanti da esecuzioni precedenti. L'azione di reinizializzazione garantisce una partenza pulita.

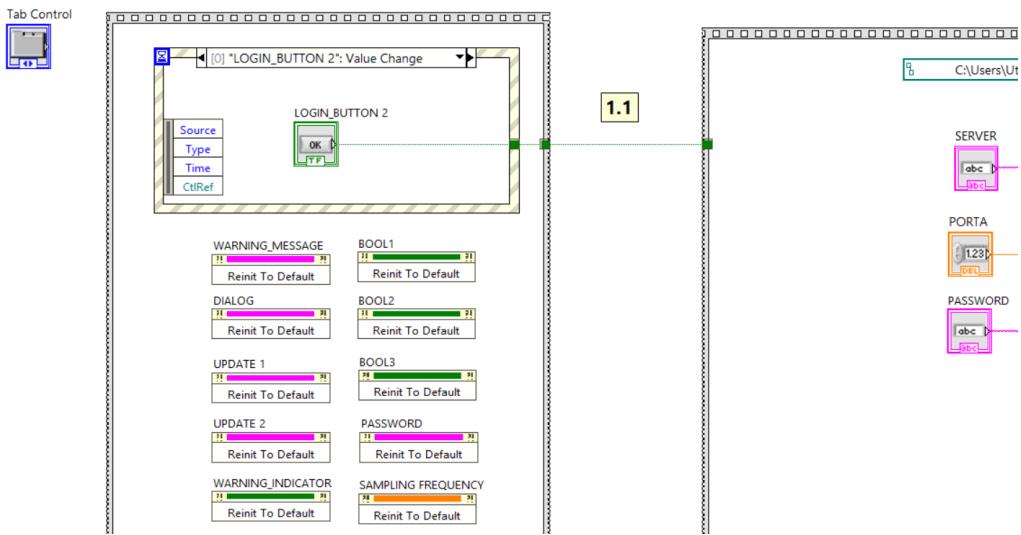


Figura 15: Block Diagram della struttura LabVIEW “Event”

Dopo aver premuto il pulsante di LOG IN nel frame della scheda di LOG IN, i valori inseriti dall'utente per i controlli di SERVER, PORTA e PASSWORD vengono attivati. Successivamente, il valore della PASSWORD viene sottoposto a un processo di crittografia delle password attraverso un collegamento con Python 3.9. Questo collegamento è reso possibile da una funzione LabVIEW denominata “*Python Node*”, che consente di richiamare direttamente funzioni Python specifiche (vedi Figura 16).

La funzione di crittografia implementata in Python è denominata “*cryptString*” e utilizza la libreria *hashlib* per eseguire la crittografia SHA-256 del valore passato come parametro. Il risultato della crittografia è quindi convertito in formato esadecimale prima di essere restituito.

La funzione Python Node è accompagnata da due ulteriori blocchi: “Open Python Session” e “Close Python Session”. Questi blocchi svolgono rispettivamente la funzione di aprire e chiudere la connessione con Python.

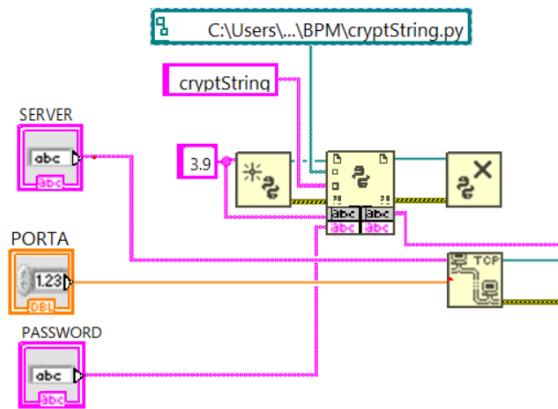


Figura 16: Block Diagram della struttura LabVIEW “Python Node” per il criptaggio della password

Come è possibile vedere in Figura 17 la password crittografata viene inviata al server per la verifica attraverso il subVI “TCP Write”, ma prima di ciò è necessario aprire una connessione tramite “TCP Open Connection”. Una volta aperta la connessione, la password crittografata viene trasmessa al server. Il server, a sua volta, riceve la password crittografata e verifica se il valore crittografato ottenuto dall'utente corrisponde alla password crittografata verificata. Se la verifica ha esito positivo, il server invia al client il messaggio “*Connection set*” (indicatore stringa DIALOG) tramite il subVI “TCP Read” precedentemente descritto. Nel caso dovesse presentarsi un errore nella fase di lettura questo viene segnalato tramite indicatore booleano *round LED* BOOL1.

Dal lato client, questo messaggio viene utilizzato come valore di “Selection label” in una struttura di *case* di LabVIEW. Solo in caso di esito positivo della verifica della password, si accede al frame del case con label “connection set”. In questo frame è possibile procedere al passaggio alla scheda HOME, consentendo all'utente di configurare i parametri di misurazione solo se la password è stata inserita correttamente. Questa implementazione assicura che l'utente possa avanzare alla HOME PAGE solo in caso di autenticazione corretta, garantendo un accesso sicuro e controllato alle funzionalità successive del programma.

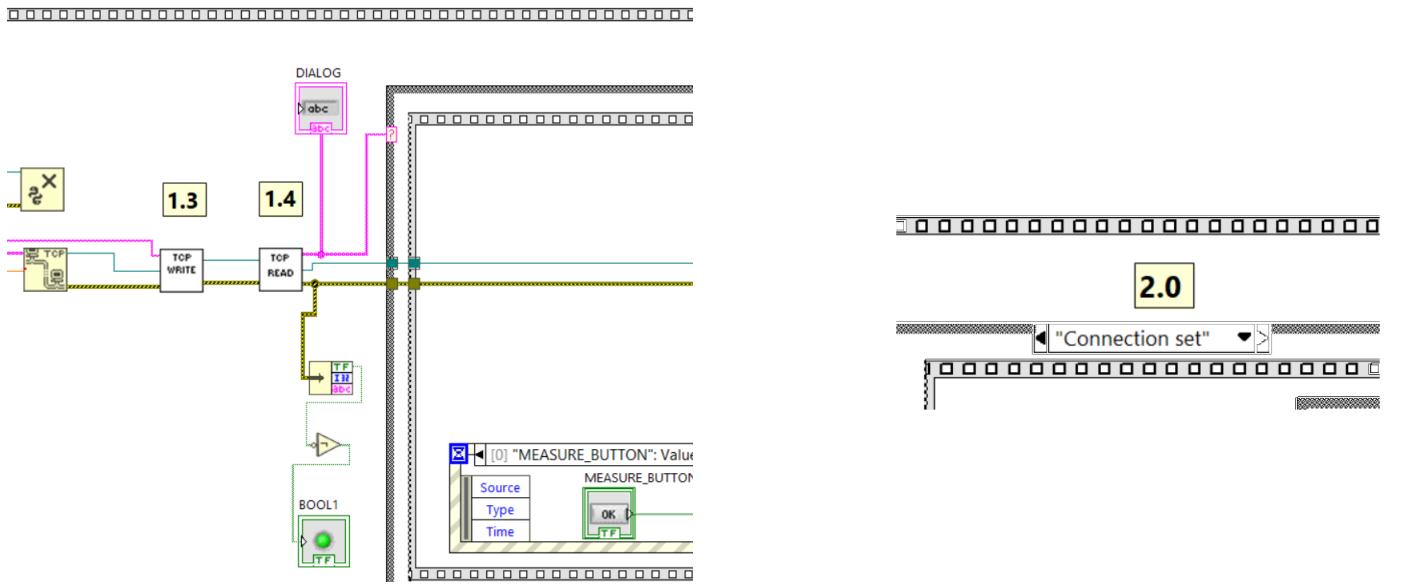


Figura 17: Elementi della parte del Block Diagram relativa alla ricezione di una risposta dal server per sapere se la connessione è stata effettivamente stabilita o meno attraverso il messaggio “Connection set”

Come è possibile vedere in Figura 18, una volta all’interno della scheda HOME PAGE, è stata implementata nuovamente la struttura “EVENT”, all’interno della quale è posizionato il pulsante “MEASURE”. Questo impedisce che l’esecuzione proceda oltre fino a quando il pulsante non viene premuto. All’interno del frame “case” più esterno relativo alla scheda HOME PAGE, è stata introdotta un’ulteriore sequenza di frame “Sequence”.

Attraverso questa struttura, nei primi due frame, i valori inseriti nei controlli relativi ai “SIGNAL PARAMETERS” (SAMPLING FREQUENCY, OBSERVATION TIME, MIN BPM e MAX BPM) non vengono elaborati fino a quando il tasto “MEASURE\_BUTTON” non viene premuto. Questo approccio consente di inviare al server i parametri di misurazione solo quando l’utente ha confermato l’intenzione di avviare la misurazione premendo il pulsante “MEASURE\_BUTTON”.

Una volta che il tasto “MEASURE” è stato premuto, i valori di SAMPLING FREQUENCY e OBSERVATION TIME vengono convertiti in stringa utilizzando il convertitore LabVIEW “Number/String Conversion” denominato “Number to Fractional String”, con una precisione di 6 (numero di cifre dopo la virgola decimale). Dopo questa conversione, ciascun parametro viene inviato al server attraverso il subVI “TCP Write”.

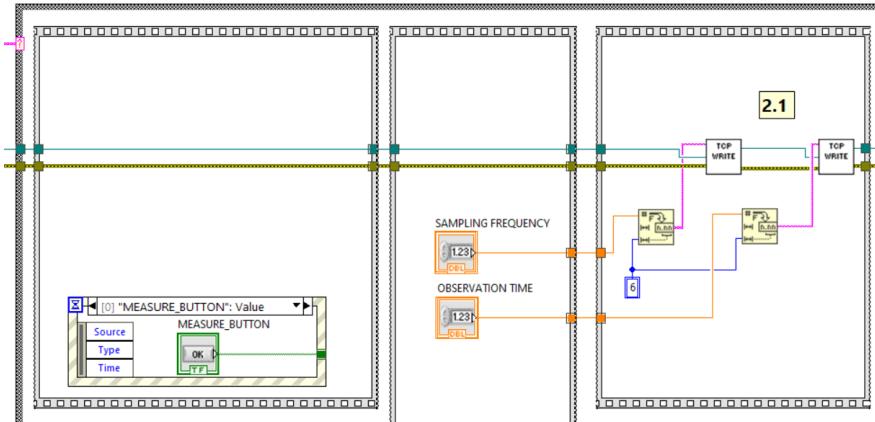


Figura 18: Parte del Block Diagram relativa all'invio al server dei SIGNAL PARAMETERS (frequenza di campionamento, Tempo di osservazione) scelti dal client premendo il pulsante “MEASURE\_BUTTON”.

Si veda adesso la Figura 19. Una volta inviati i parametri di misurazione, il client attende che il SERVER esegua l'acquisizione del segnale e deve essere informato sull'evoluzione del processo. Utilizzando il subVI “TCP Read”, il client riceve un messaggio dal server che segnala l'inizio della misurazione, identificato dal messaggio “UPDATE 1” e dal corrispondente indicatore booleano LED “BOOL2”.

Attraverso un'ulteriore struttura “Case” di LabVIEW, il client procede a ricevere informazioni sul segnale solo se durante il processo di acquisizione non si sono verificati errori. In questo caso, il subVI “TCP Read” con l'indicatore “UPDATE 1” restituisce il messaggio “Starting measure”. Successivamente, si accede al frame di case di default, dove il client riceve il segnale stesso tramite il subVI “Read WFM” e ottiene la segnalazione di fine misurazione con il messaggio “Measure terminated” attraverso l'indicatore stringa “UPDATE2” e il corrispondente indicatore booleano LED “BOOL3”. Se durante l'acquisizione si riscontra qualche problema, il flusso del programma cade nel caso “ERROR” della struttura di case, il quale rappresenta il messaggio di errore inviato al client tramite il subVI “TCP READ” con indicatore “UPDATE 1”. Questo porta direttamente alla chiusura della connessione TCP con il server attraverso il blocco TCP “Close Connection”, evitando di procedere alla parte di elaborazione del segnale e al calcolo del valore di pulsazione cardiaca.

Dopo aver ricevuto il segnale di pulsazione, questo viene visualizzato tramite un componente “Waveform Graph” sul Front Panel. Simultaneamente, il segnale viene inviato al subVI “BPM” per essere elaborato e ottenere il valore di pulsazione cardiaca in battiti per minuto (BPM). Questo valore elaborato viene restituito sotto forma di indicatore “HEART RATE [BPM]”, previo arrotondamento da tipo double a intero mediante il convertitore LabVIEW “To Long Integer I32”.

Inoltre, il segnale in output dal subVI “BPM” — quindi il segnale elaborato e processato — viene nuovamente visualizzato sull’interfaccia utente utilizzando altri due “Waveform Graph”. Uno di essi mostra il segnale processato nel dominio del tempo con l’etichetta “PROCESSED SIGNAL”, mentre l’altro, denominato “PROCESSED FFT-MAG”, rappresenta il segnale nel dominio della frequenza.

Quest'ultimo grafico è ottenuto applicando la funzione di elaborazione del segnale “FFT Spectrum (Mag-Phase)” al segnale processato, consentendo la visualizzazione della rappresentazione FFT nel dominio della frequenza. Questa implementazione offre una completa rappresentazione grafica del segnale di pulsazione e del suo spettro di frequenza.

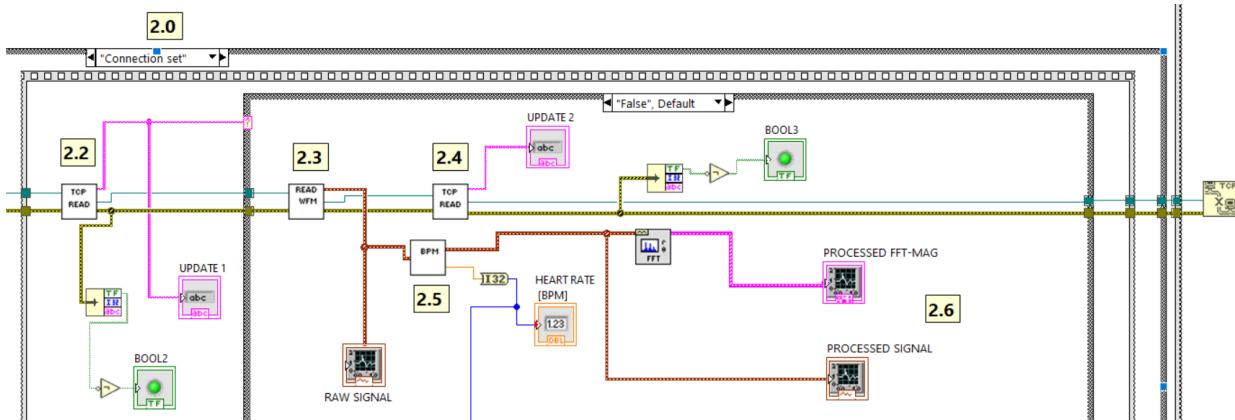


Figura 19: Parte del Block Diagram relativa all'elaborazione del segnale ricevuto dal server, al calcolo del valore di pulsazione cardiaca in BPM e alla visualizzazione grafica dei risultati.

Nel diagramma a blocchi in Figura 20, dopo aver ottenuto il valore della pulsazione cardiaca, viene eseguita un'analisi aggiuntiva per classificarlo come NORMAL, TACHICARDICO o BRADICARDICO, in base ai limiti imposti dall'utente tramite i controlli “BPM MIN” e “BPM MAX”. Per realizzare ciò, sono impiegate le funzioni di confronto di LabVIEW (“Greater or equal?” e “Less or equal?”), nonché gli operatori booleani AND e NOT, insieme a strutture di controllo *Case* per definire le diverse condizioni.

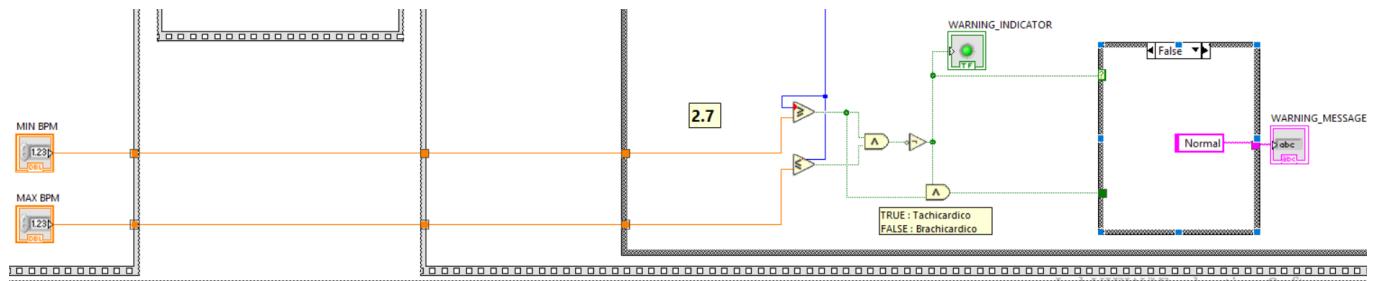


Figura 20: Parte del Block Diagram che permette di verificare se il valore del battito misurato appartiene all'intervallo delimitato dai livelli limite imposti dal cliente.

La funzione “*Greater or equal?*” prende come input il valore della pulsazione cardiaca HEART RATE [BPM] e il valore imposto BPM MIN, mentre “*Less or equal?*” utilizza anch’essa il valore misurato della pulsazione ma confrontato con BPM MAX. I risultati di entrambi i confronti sono passati attraverso l’operatore AND seguito da NOT, in modo che se il risultato è FALSE, si proceda al caso nel quale l’indicatore di avviso restituisce la stringa “*Normal*”. L’output del NOT e quello del “*Greater or equal?*” sono collegati a un altro AND, in modo che, rimanendo nel caso TRUE del primo *case* (quando la pulsazione non è tra i due limiti), se la pulsazione è maggiore del limite minimo (si veda Figura 21), si accede al caso nel quale il messaggio dell’indicatore WARNING\_MESSAGE sarà “*Tachicardico*” (frame TRUE del *case* annidato nel frame TRUE del *case* precedente). In caso contrario (si veda Figura 22), il messaggio sarà “*Brachicardico*” (frame FALSE del *case* annidato nel frame TRUE del primo *case*).

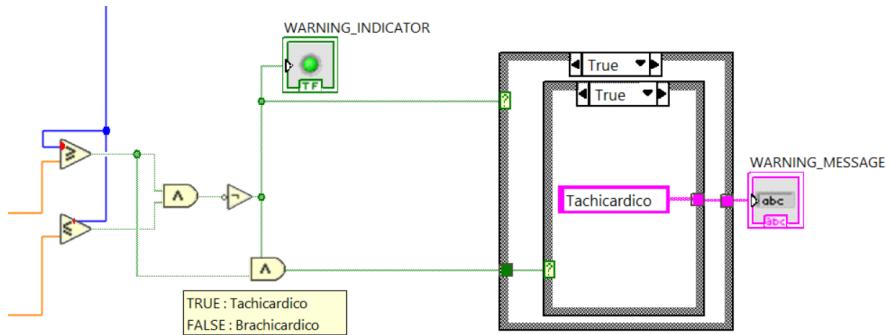


Figura 21: Parte del Block Diagram che permette di verificare se il valore del battito misurato risulti maggiore del limite massimo imposto e quindi se il paziente sia *Tachicardico*.

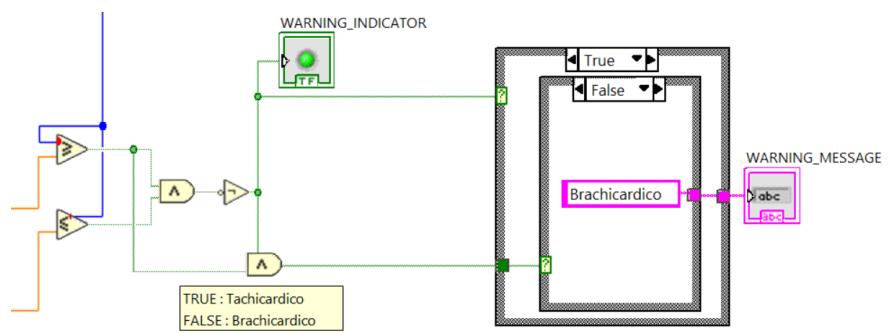


Figura 22: Parte del Block Diagram che permette di verificare se il valore del battito misurato risulti minore del limite minimo imposto e quindi se il paziente sia *Brachicardico*.

## 4 Server

L'applicazione presentata in questo documento risulta essere modulare: permette lo sviluppo parallelo dei vari moduli, dal lato client e da quello server, mantenendo, chiaramente, le stesse regole di comunicazione. Tale condizione promuove la possibilità di modifica in ciascuno dei moduli senza la necessità di intervenire sugli altri.

L'architettura in cui il client richiede un'azione o un servizio da parte di un fornitore, si completa con l'implementazione di quest'ultimo. Il server può garantire un'esecuzione unitaria (“run”) oppure continua (“run continuously”) in ambiente di programmazione LabVIEW tramite gli opportuni comandi, che permette al client di richiedere servizi in un qualsiasi istante temporale. Così come esposto nel lato client, anche nel lato server è possibile visualizzare una struttura sequenziale caratterizzata dal flusso di dati garantito dalla presenza dei collegamenti tra “connection ID” e “error”. Il server si occupa dell'acquisizione dei dati provenienti del sensore e della trasmissione di questi al client per il processing e la visualizzazione.

Il server si predisponde, quindi, alla rilevazione delle richieste provenienti dal client per poi soddisfarle attraverso il processo. La passività dell'apparato in questione si denota a seguito dalle specifiche progettuali per la realizzazione del server. Si richiede, infatti, che questo sia in grado di gestire la richiesta di un client predisponendosi in ascolto, riconoscendone l'autenticità, autorizzandone l'accesso e consentendogli l'impostazione di parametri specifici di interesse medico, caratteristici del segnale in acquisizione (frequenza di campionamento e tempo di esecuzione).

Deve essere, per cui, efficace nell'avvio della misurazione segnalando, contestualmente, l'evento al client che deve, altresì, avere modo di individuarne il termine. Infine, al server, viene demandato il compito di inviare i risultati al client da cui erano stati precedentemente richiesti.

Le suddette specifiche sono state soddisfatte e risolte tramite la progettazione di un Virtual Instrument di cui, di seguito, vengono riportate struttura ed entità. Così come discusso in 3.1, anche il lato server presenta due sezioni distinte dallo scopo di chiarire le modalità di funzionamento e la logica di progettazione, riportando la sezione della documentazione del software lato utente e programmatore.

## 4.1 SERVER: Manuale utente

L'interfaccia proposta è progettata per consentire all'utente di interagire con il programma in modo intuitivo, mostrando la possibilità di controllare strumenti e dispositivi per assicurare la visualizzazione di dati in tempo reale. Il Front Panel riportato in Figura 23 mostra la struttura proposta che consta di diversi elementi di seguito descritti nel dettaglio:

- PORT: Controllo in grado di consentire la comunicazione tra server e client che necessita dell'impostazione del numero di porta in modo che esso sia corrisposto dal lato client.
- SAMPLING FREQUENCY
- OBSERVATION TIME

Gli indicatori “SAMPLING FREQUENCY” ed “OBSERVATION TIME” permettono all'utente di avere informazioni accessibili e dirette sull'impostazione che viene disposta, da parte del client, rispettivamente, della frequenza di campionamento del segnale e del tempo di esecuzione della misura.

- # POINTS: Indicatore che mostra il numero di campioni corrispondente al prodotto tra la frequenza di campionamento e il tempo di osservazione.
- DAQ CHANNEL : Controllo che permette di selezionare il canale per assicurare la comunicazione con il dispositivo DAQ (2).

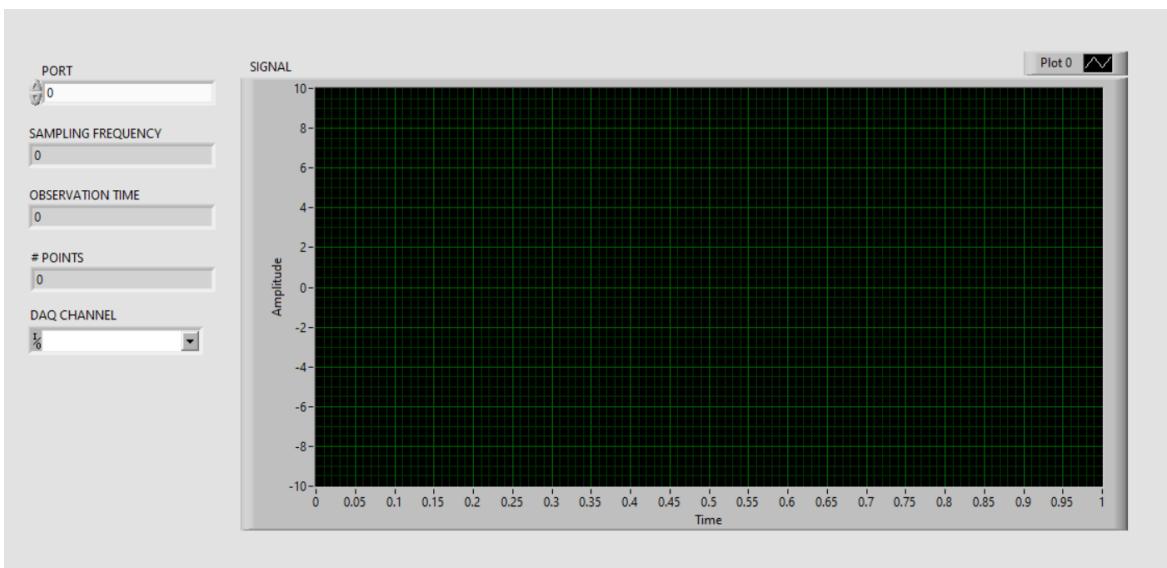


Figura 23: Interfaccia del server

## 4.2 SERVER: Manuale programmatore

La seguente sezione si propone di presentare il Block Diagram, elemento fondamentale della programmazione visuale nell'ambiente LabVIEW, al fine di metterne in risalto la logica di esecuzione, l'insieme di funzionalità e la coerenza di sviluppo per garantire all'utente stabilità ed affidabilità in linea con i requisiti funzionali specificati in fase progettuale.

Ci si propone, quindi, di presentare un'implementazione che favorisca una performance efficiente adeguatamente testata, garantisca sicurezza nella gestione di dati sensibili (Figura 24).

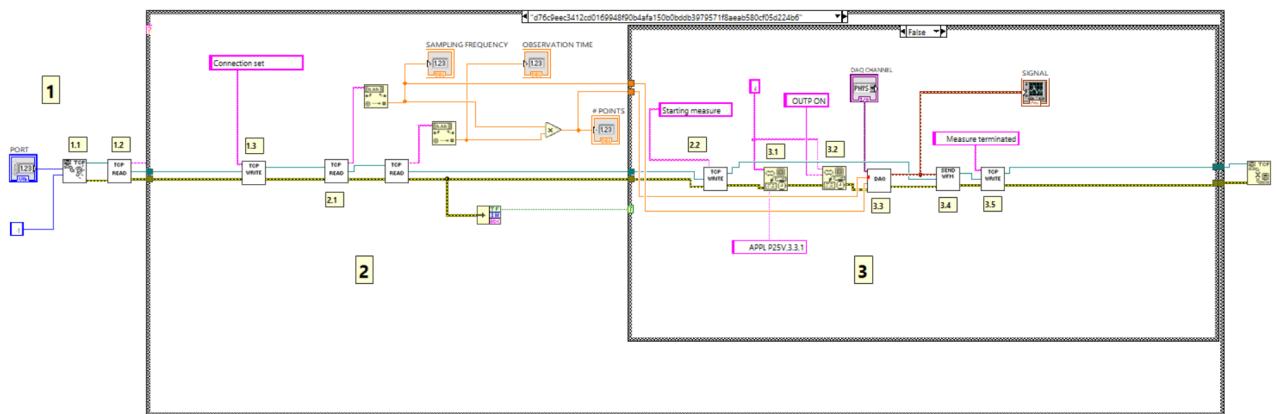


Figura 24: Struttura del Block Diagram del server

A tal proposito si presentano, di seguito, le varie sezioni che compongono il Block Diagram ponendo l'attenzione sulle differenti fasi del processo di misurazione della frequenza cardiaca. Mantenendo l'impostazione inizializzata nella sezione 3.2, è possibile descrivere il Block Diagram in tre zone distinte che si vanno a posizionare ad incastro nel flusso logico ivi presentato.

La prima zona (Figura 25) riguarda l'interazione con il client nel processo di autenticazione, la seconda (Figura 26) la ricezione delle impostazioni sui parametri di misura e la terza (Figura 27), infine, riguarda l'interazione con la strumentazione e la successiva trasmissione del segnale acquisito.

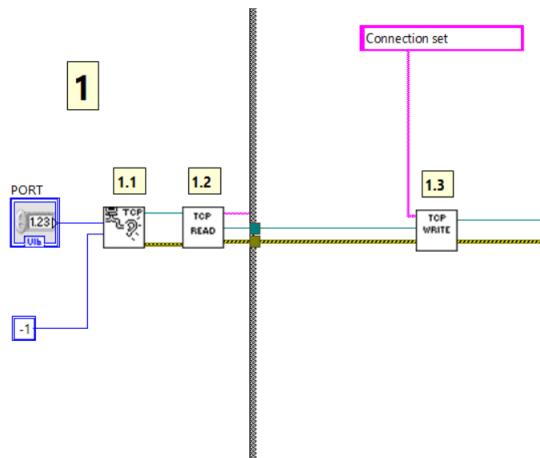


Figura 25: Block Diagram relativo alla zona di "LOG IN AND CONNECTION"

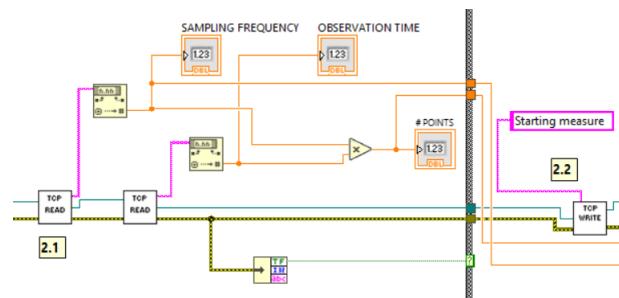


Figura 26: Block Diagram relativo alla zona di "SIGNAL PARAMETERS"

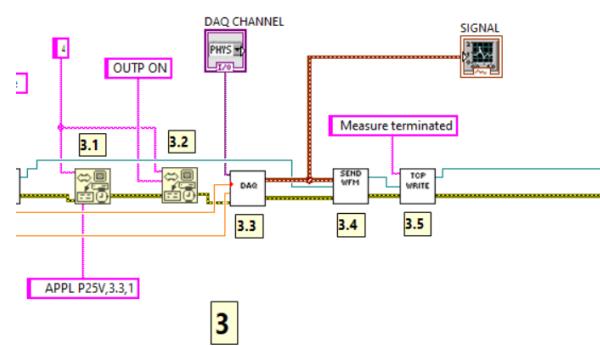


Figura 27: Block Diagram relativo alla zona di "INTERACTION WITH MEASUREMENT EQUIPMENT"

Anche in questo caso le varie fasi del flusso operativo sono state enumerate.

Di seguito vengono riportati i passaggi logici eseguiti nel Block Diagram (Figure 28, 29, 30)

### 1. LOG IN AND CONNECTION

- 1.1 *Creating a liststener*
- 1.2 *Receiving the encrypted password by the client*
- 1.3 *Verifying password and sending a message of connection set*

Figura 28: Fasi del flusso operativo relative alla scheda di "LOG IN AND CONNECTION"

### 2. SIGNAL PARAMETERS

- 2.1 *Acquiring the SIGNAL PARAMETERS selected by the client*
- 2.2 *Sending the measurement start message to the client*

Figura 29: Fasi del flusso operativo relative alla scheda di "SIGNAL PARAMETERS"

### 3. INTERACTION WITH MEASUREMENT EQUIPMENT

- 3.1 *Sending a device message to HP Agilent Keysight E3631A*
- 3.2 *Enabling of outputs of HP Agilent Keysight E3631A*
- 3.3 *Acquiring data with DAQ*
- 3.4 *Sending the acquired heart rate pulse signal*
- 3.5 *Sending the measurement termination message*

Figura 30: Fasi del flusso operativo relative alla scheda di "INTERACTION WITH MEASUREMENT EQUIPMENT"

Anche nel caso del server vengono utilizzati molteplici **subVI** come componenti chiave dai quali è dipesa l'organizzazione, la chiarezza e la modularità del codice. Di seguito vengono riportati quelli introdotti nel Block Diagram presentato:

- *TCP Write;*
- *TCP Read;*
- *DAQ;*
- *Send WFM;*

La descrizione dei primi due SubVI è riportata in 3.2.

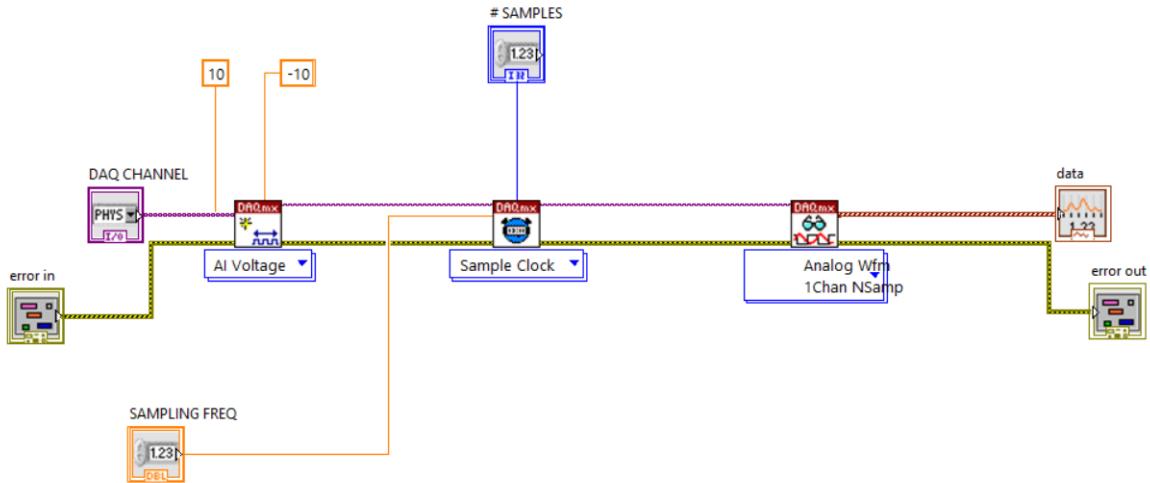


Figura 31: Block Diagram del subVI “DAQ”

Il SubVI *DAQ* mostrato in Figura 31 è caratterizzato da un insieme di funzioni che permettono di gestire il sistema di acquisizione dati con una scheda DAQ.

Questo è costituito da tre blocchi principali: “*DAQmx Create Channel*”, “*Sample Clock*” e “*DAQmx Read*”. Il primo crea un canale per l’acquisizione del segnale, il secondo permette di gestire il tempo di osservazione del fenomeno, la frequenza di campionamento e il numero di campioni per canale.

Il blocco “*DAQmx Read*”, infine, riceve come input le informazioni di configurazione del canale provenienti dal DAQ CHANNEL poichè costituiscono parametri necessari per la lettura di dati dal dispositivo e restituisce i dati letti sotto forma di segnale. L’utilizzo di questo blocco consente di integrare facilmente la lettura di dati da dispositivi di acquisizione dati NI-DAQmx nei programmi LabVIEW. Questo è particolarmente utile in applicazioni che coinvolgono misurazioni e acquisizioni di segnali provenienti da sensori o strumenti di misura.

Nello specifico, “*DAQmx Create Channel*” prende in ingresso i valori +10 e -10 definiti come parametri di fondo scala. Questi rappresentano l’intervallo massimo o la massima estensione di misura dello strumento e sono specifici per ciascun canale di ingresso, indicando l’intervallo di valori che il dispositivo può misurare accuratamente. Nel caso in analisi si ha un fondo scala di  $\pm 10$  V decretando la possibilità di misurare con precisione tensioni comprese tra -10 V e +10 V.

Se il segnale di ingresso dovesse superare questi limiti, potrebbe verificarsi la saturazione e il dispositivo potrebbe non essere in grado di fornire una misura accurata. Ponendo l’attenzione sul blocco “*DAQmx Read*” si può notare l’impostazione “*Analog Wfm 1Chan NSamp*” che sta ad indicare l’utilizzo di un unico canale analogico (nel caso presentato corrisponde allo 0) per l’acquisizione di un numero multiplo di campioni. Il collegamento in differenziale fa sì che la DAQ misuri la differenza di tensione tra questi due punti anziché rispetto a un punto di riferimento comune. Questo approccio è utile per ridurre i disturbi e i rumori che possono essere introdotti da segnali comuni a entrambi i canali. Per questa ragione, si è ritenuto opportuno lavorare in modalità differenziale andando a collegare il “*PURPLE wire*” del Pulse Sensor attraverso il quale viene trasmesso il segnale, al pin 0 (+) del canale 0 e il “*BLACK wire*” che rappresenta il GROUND in comune con l’alimentatore al pin 4 (-) dello stesso canale.

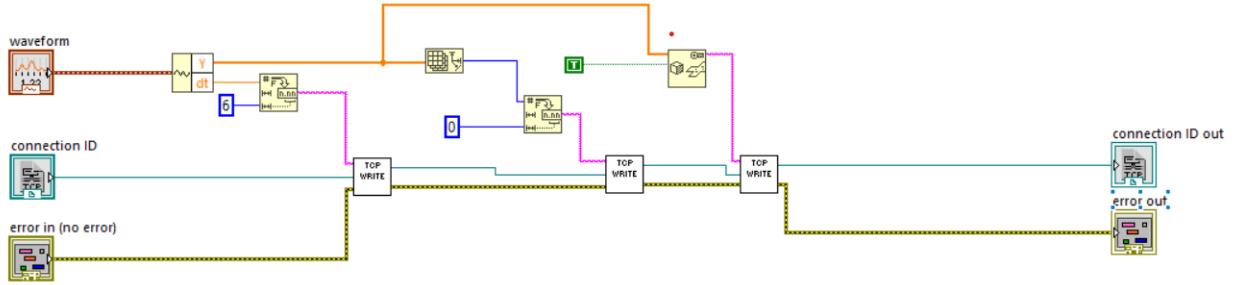


Figura 32: Block Diagram del subVI “Send WFM”

Il SubVI *Send WFM* (Figura 32) è progettato per effettuare la trasmissione al cliente del segnale di pulsazione cardiaca acquisito dal server.

È composto da tre SubVI *TCP Write* distinti. Il segnale in formato *Waveform* viene sottoposto al blocco “Get Waveform Components” che scomponete le componenti del segnale in “data value” e “time interval” che saranno presi in ingresso rispettivamente dal primo e dal terzo *TCP Write*. In ingresso, in aggiunta, ognuno dei tre acquisisce *Connection ID* ed *Error in* e come output presenta *Connection ID* ed *Error out*.

Il secondo, nello specifico, ha il compito di preparare il client a ricevere un certo numero di dati corrispondente al numero degli elementi dell’array - calcolato con il blocco *Array Size* – costruito sulla base dei campioni del segnale in ingresso. Si sottolinea la presenza della costante 0 per indicare la precisione relativa al convertitore che interagisce con il secondo *TCP Write* in quanto rappresentativo del numero dei campioni.

La precisione è definita dal numero di cifre dopo la virgola - nel primo caso presentato corrispondente a 6 - ed è impostata in ingresso al blocco *Number To Fractional String* che converte il numero in ingresso in una stringa.

La funzione *Flatten to String* converte qualsiasi dato in ingresso in una stringa.

I SubVI appena descritti si collocano nel contesto complessivo presentato in Figura 24, definito attraverso la composizione di due *Case Structure* ossia strutture che permettono l'esecuzione delle funzioni definite al loro interno solo nel caso in cui le condizioni imposte nella struttura in fase di progettazione siano soddisfatte dai valori in ingresso. Nella fattispecie la struttura più esterna impone la condizione di corrispondenza tra la password inserita dall'utente client e quella crittografata impostata come condizione “true” che permette, cioè, all'utente di accedere al processo di misura e, contestualmente, al server di eseguire il codice ad esso corrispondente, come riportato in Figura 33.

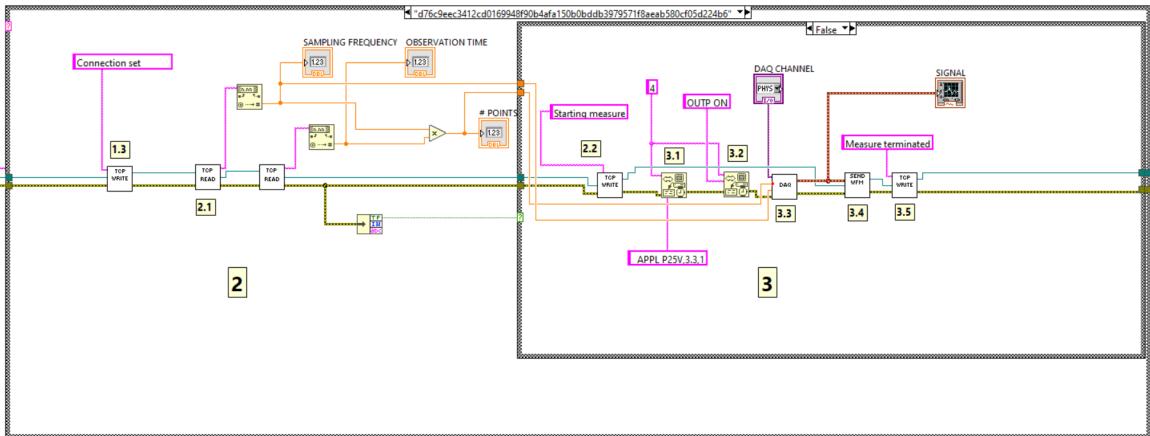


Figura 33: Block Diagram relativo alla Case Structure esterna - caso True

In caso di mancato matching, l'utente riceverà un messaggio di *Connection Refused* e il server procederà con la chiusura della connessione come azione finale (Figura 34).

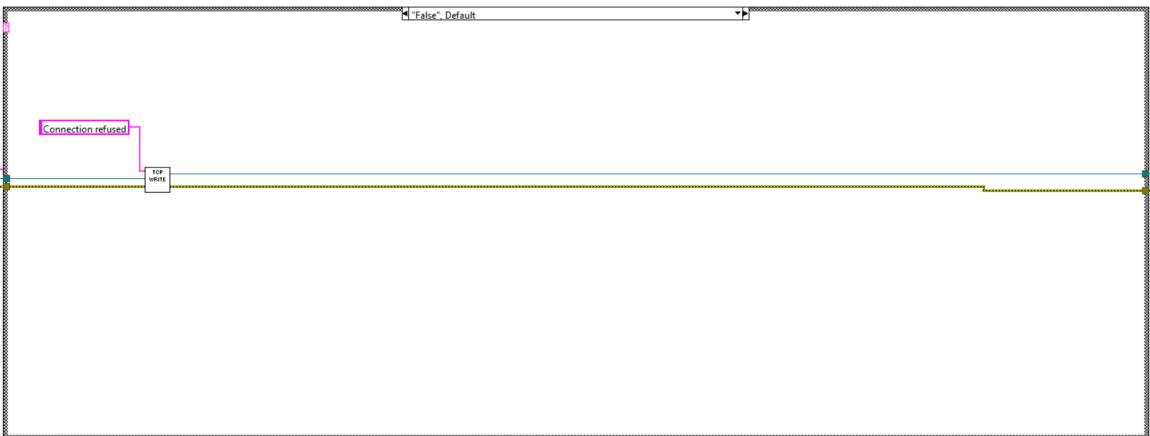


Figura 34: Block Diagram relativo alla Case Structure esterna - caso False

La seconda *Case Structure*, nel caso in cui non siano riscontrati errori, è visibile in Figura 35. In caso contrario il server provvederà all'invio di un messaggio di errore al client, terminando il processo. Inizialmente il server attiva il processo di connessione tramite l'utilizzo del protocollo TCP/IP. In primo luogo si assiste alla creazione di un ascoltatore su una porta specificata e si attende la richiesta di connessione da parte del client per un tempo indeterminato impostato tramite la costante -1.

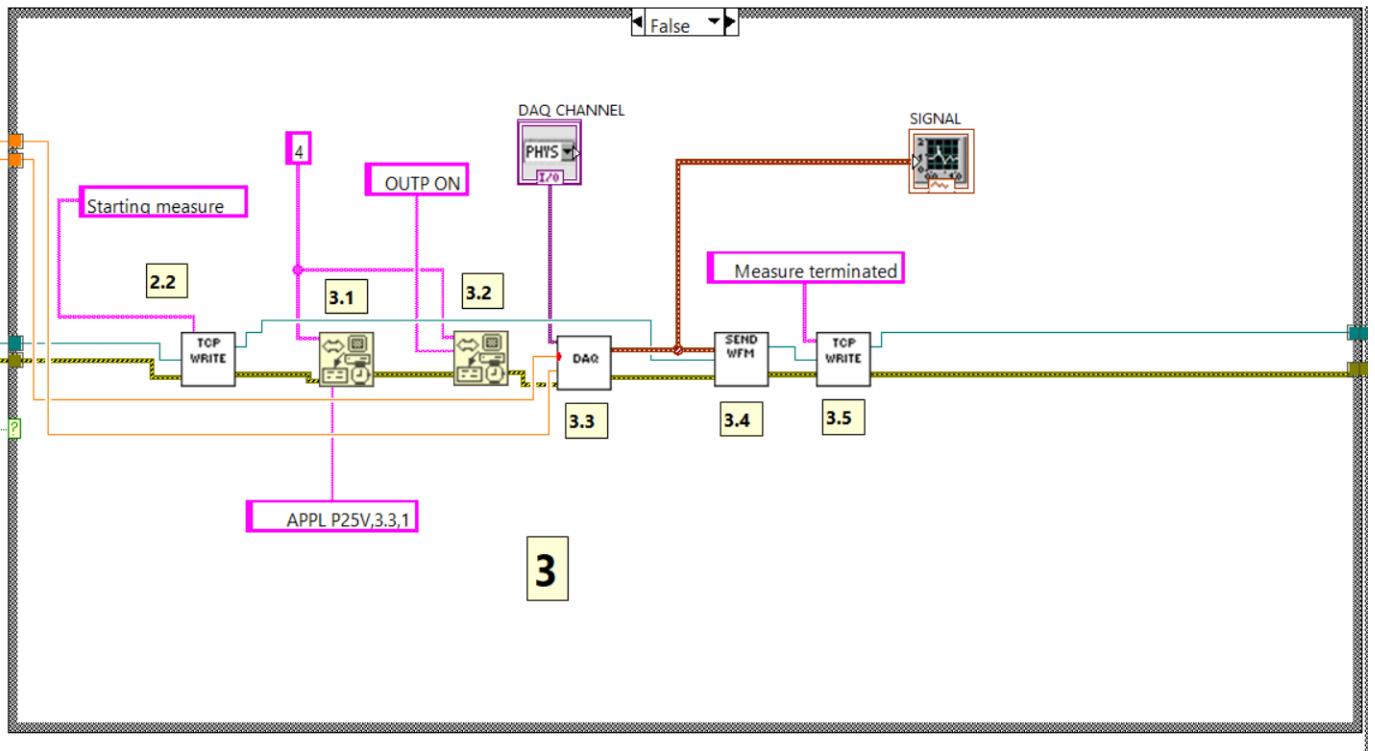


Figura 35: Block Diagram relativo alla Case Structure interna

Il SubVI *TCP Read* permette la connessione e la successiva trasmissione della password crittografata da parte del client. Essendo, questa, condizione positiva della prima *Case Structure*, viene verificata e il *TCP Write* permette la trasmissione del messaggio di connessione effettuata al client.

Dal successivo *TCP Read* il server acquisisce i valori di frequenza di campionamento e tempo di osservazione. In ciascuno dei due si nota la presenza della funzione *Fract/Exp String To Number* che si propone di convertire il dato trasmesso dal client dal formato stringa al formato numerico.

Questi valori vengono, successivamente, condotti all'interno della seconda struttura case nella quale, dopo il primo *TCP Write* con cui il client viene messo al corrente dell'inizio della misura, sono presenti due blocchi *GPIB Write*.

La funzione *GPIB Write* in LabVIEW è specificamente progettata per interagire con dispositivi GPIB ed è utilizzata per scrivere dati o comandi su un dispositivo collegato attraverso l'interfaccia GPIB (General Purpose Interface Bus).

Questa funzione prende in ingresso l'Indirizzo dello strumento (4), la stringa di comando vero e proprio che rispetti la sintassi definita nel manuale dello strumento e l'*error in*. Nel primo caso il commando *APPL P25V,3.3,1* è stato utilizzato per comunicare al generatore HP Agilent Keysight E3631A l'impostazione di un intervallo di tensione di uscita +25V ad un valore di 3.3V, 1A. Nel secondo caso *OUTP ON* è stato necessario per abilitare le uscite. (Si veda Figura 36). Successivamente si nota il SubVI *DAQ* che produce, in uscita, il segnale visualizzabile dal Front Panel, che verrà, poi, restituito al client attraverso il seguente SubVI *Send WFM*. L'ultimo blocco *TCP Write* è necessario per l'invio di un messaggio di misura terminata al client. Il codice termina chiudendo la connessione.

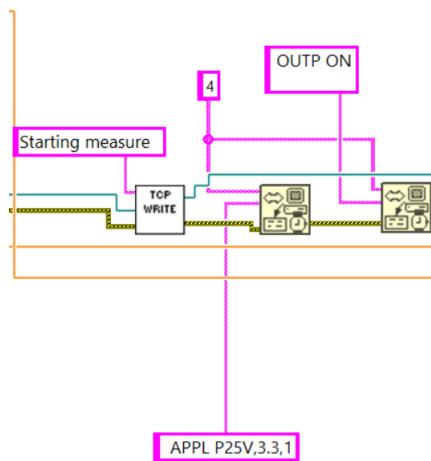


Figura 36: Block Diagram relativo alla comunicazione con lo strumento

## 5 Esempio di funzionamento e bug riscontrati

### 5.1 Esempio di funzionamento

Per valutare l'efficacia del sistema di misura progettato e implementato su LabVIEW per la misurazione della pulsazione cardiaca, è stato condotto un test di caratterizzazione in ambiente sperimentale dal vivo. L'obiettivo principale è quello di verificare se il set up sperimentale e l'implementazione dei vari blocchi nell'ambiente di programmazione LabVIEW fossero in grado di fornire una visualizzazione di segnale valida e di misurare accuratamente la pulsazione cardiaca di un individuo utilizzando un pulsossimetro.

Nell'illustrare (Figure 37-38) il funzionamento del sistema progettato, la strumentazione è stata accuratamente collegata seguendo le specifiche delineate nel set up di misura (2.1).



Figura 37: L'HP Agilent Keysight E3631A permette di alimentare il sensore PulseSensor.

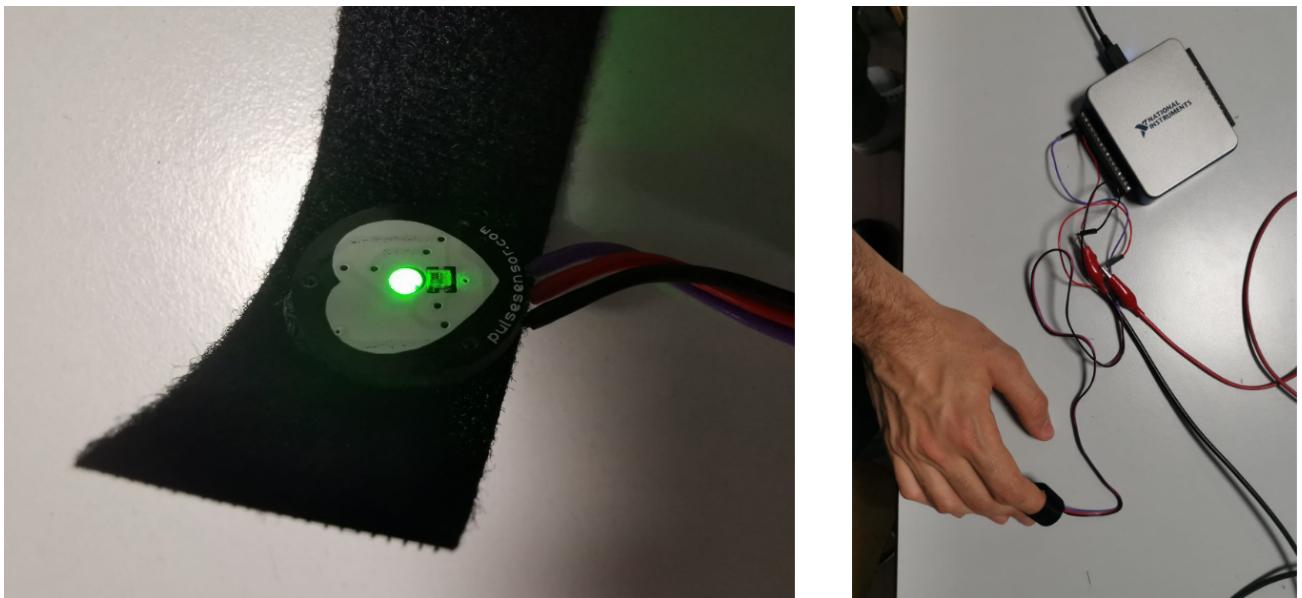


Figura 38: *Pulse Sensor* misura il segnale dal paziente e la scheda di acquisizione (*NI DAQmx USB-6001*) ne effettua l'acquisizione



Figura 39: Il Server tramite interfaccia USB-GPIB permette di pilotare l'alimentatore e tramite un'altra porta USB riceve il segnale acquisito dalla scheda DAQ per poi inviarlo al Client secondo protocollo TCP/IP

Visualizzando il Front Panel dal lato client (Figura 40) nella scheda di LOG IN esso si collega con successo al server inserendo accuratamente l'indirizzo IP (192.168.43.6) e il numero di porta (1000), come specificato nell'esempio in questione. È fondamentale fornire la password corretta per autorizzare la connessione. Una volta che questi parametri sono stati inseriti correttamente, il client riceve immediatamente la conferma della connessione attraverso il messaggio "Connection Set".

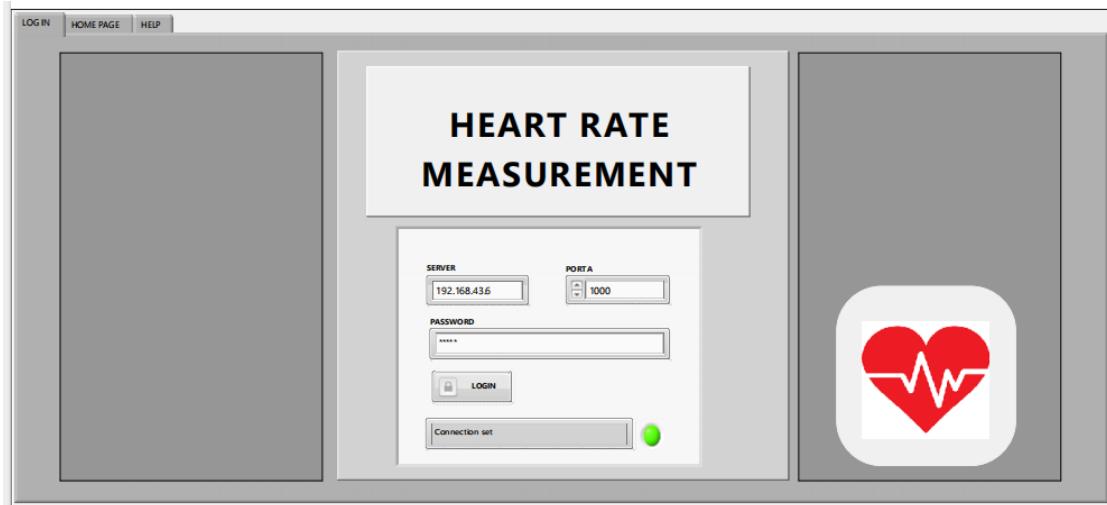


Figura 40: Front Panel del Client - Scheda LOG IN

Sul Front Panel del lato server (Figura 41), è visibile che il numero di porta coincide con quello configurato sul lato client, fornendo un riferimento immediato per garantire una corretta corrispondenza. Inoltre, vengono visualizzati come indicatori i parametri di segnale impostati dal client, offrendo una chiara rappresentazione delle specifiche di configurazione stabilite. Il pannello frontale del server mostra in maniera dettagliata il segnale grezzo acquisito dalla scheda DAQ, consentendo una valutazione diretta delle informazioni provenienti dal sistema di acquisizione.

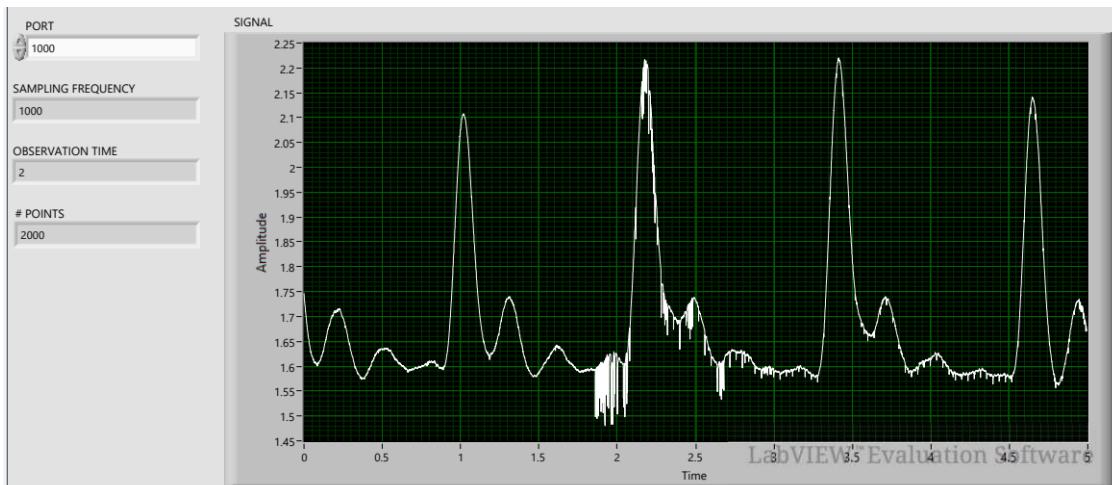


Figura 41: Front Panel del Server

Infine, sulla scheda HOME PAGE del Front Panel del client (Figura 42), è possibile visualizzare in basso i parametri di misura configurati dal client. Questi parametri hanno guidato la generazione dei grafici nel dominio del tempo e della frequenza del segnale di pulsazione cardiaca. Particolare attenzione è rivolta al processo di elaborazione del segnale implementato, il quale ha notevolmente migliorato la qualità della traccia rispetto al segnale grezzo originariamente acquisito. Inoltre, viene riportato il valore della frequenza cardiaca (50 BPM), con un avviso specifico che indica il superamento dei limiti imposti dal client. Questo dato riflette una condizione brachicardica del paziente, sottolineando la capacità del sistema di rilevare e segnalare eventuali anomalie nella frequenza cardiaca sulla base dei valori limite impostati dall'utente.

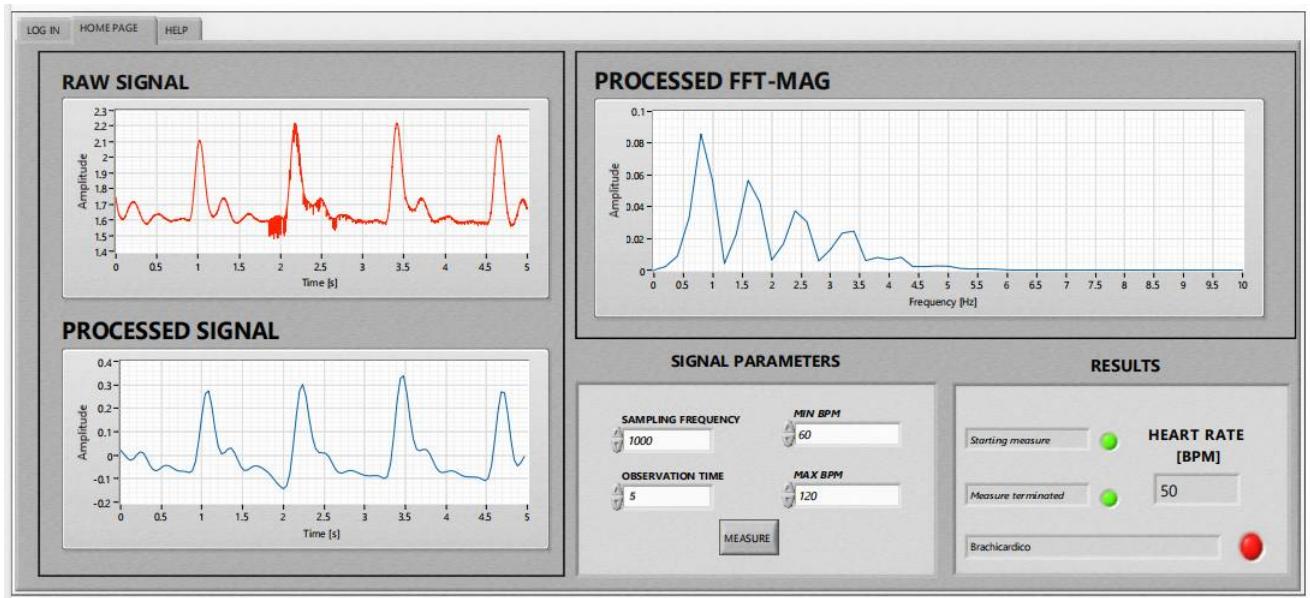


Figura 42: Front Panel del Client - Scheda HOME PAGE

## 5.2 Bug riscontrati

Di seguito vengono elencati i bug di sistema identificati e irrisolti.

- L'interruzione anomala dell'applicativo SERVER comporta l'interruzione dell'applicativo CLIENT. In particolare, se l'arresto avviene in fase di invio del segnale acquisito verrà generato un messaggio di errore dall'ambiente LabVIEW relativo alla presenza di dati corrotti.
- Impostare i valori *MAX BPM* e *MIN BPM* in modo tale da avere *MIN BPM* > *MAX BPM* potrebbe provocare una falsa attivazione del segnale di warning anche in condizioni di battito normale.
- Problemi di trasferimento dati possono avvenire utilizzando connessioni non stabili o utilizzando la modalità LOCALHOST. La causa di quest'ultimo problema non è stata ancora individuata.