

INTRODUCING

MYOWARE 2.0

Any sufficiently advanced muscle sensor is
indistinguishable from myoelectric magic



UNIVERSITÀ DEGLI STUDI DEL MOLISE

DIPARTIMENTO DI MEDICINA E SCIENZE DELLA SALUTE

Corso di Laurea Magistrale in

INGEGNERIA BIOMEDICA

Esame di Sistemi di misura distribuiti

Realizzazione di un sistema di misura distribuito per
l'acquisizione di un segnale Elettromiografico



UNIVERSITÀ
DEGLI STUDI
DEL MOLISE



Anno accademico 2023-2024
UNIVERSITÀ DEGLI STUDI
DEL SANNIO Benevento



UNIVERSITÀ DEGLI STUDI
DI CASSINO E DEL
LAZIO MERIDIONALE

Indice

1	Introduzione	2
2	Strumentazione utilizzata	3
3	Client: manuale utente	7
3.1	Accesso al server	7
3.2	Definizione dei parametri per l'acquisizione	8
3.3	Avvio del processo di misura	8
3.4	Impostazione parametri di soglia	9
3.5	Visualizzazione Elettromiogramma	10
4	Manuale Programmatore	10
4.1	Sub-VI	10
4.2	Client: manuale programmatore	14
4.3	Server: manuale programmatore	18
5	Conclusioni	21

1 Introduzione

L'attività laboratoriale assegnata ha previsto la realizzazione di un sistema di misura distribuito in grado di acquisire un segnale elettromiografico, visualizzarlo nel dominio del tempo e stimarne i valori massimo ed efficace (RMS).

Nel progetto è stato configurato un sistema di acquisizione dati per monitorare l'attività muscolare mediante l'utilizzo di un sensore elettromiografico *Myoware Muscle Sensor AT-04-001*, alimentato tramite un *Agilent E3631A*. Il sensore ha consentito la misurazione accurata dell'attività elettrica nei muscoli durante il movimento.

Al fine di acquisire i segnali EMG, è stata utilizzata una scheda d'acquisizione dati (*DAQ*) *NI USB-6001/6002/6003*, che ha permesso la conversione dei segnali analogici in segnali digitali necessari per l'elaborazione successiva.

Per consentire il collegamento del computer all'alimentatore *Agilent E3631A* è stata adoperata un'interfaccia *GPIB IEEE-488*, la quale ha permesso il controllo remoto e la configurazione dei parametri di alimentazione.

Il software impiegato è stato *NI MAX (Measurement & Automation Explorer)* e *LABVIEW*. Tale combinazione ha consentito la progettazione di un'interfaccia utente per:

- Controllare l'alimentatore;
- Acquisire i segnali EMG della scheda DAQ;
- Analizzare i dati in tempo reale.

In tal modo è stato possibile stabilire una connessione *TCP-IP* tra un *client* ed un *server*; quest'ultima ha permesso la comunicazione dell'utente con la strumentazione impiegata per la misurazione del segnale.

L'interfaccia *client*, appare efficace ed esplicativa; essa infatti è stata sviluppata in modo da permettere all'utente, che accede al sistema mediante un processo di autorizzazione, di pilotare da remoto la strumentazione e il processo di misura. D'altro canto, l'interfaccia *server*, dopo aver riconosciuto l'utente *client* come autorizzato dalla *password*, imposta i parametri di acquisizione scelti dall'utente e da inizio alla misura.

Inoltre, l'implementazione del sistema ha previsto dal lato *client* di realizzare un segnale di *warning* nell'ipotesi in cui il valore massimo ed efficace dell'EMG risultino essere inferiori rispetto alle soglie impostabili arbitrariamente dall'utente.

2 Strumentazione utilizzata

La realizzazione del sistema di misura per l'attività elettromiografica ha coinvolto l'utilizzo dei seguenti strumenti:

a. MyoWare MUSCLE SENSOR (AT-04-001)

Myoware Muscle Sensor AT-04-001^[1], riportato in **Fig.1**, è un sensore progettato per misurare segnali elettromiografici, consentendo il monitoraggio dell'attività muscolare durante il movimento o l'esercizio fisico.

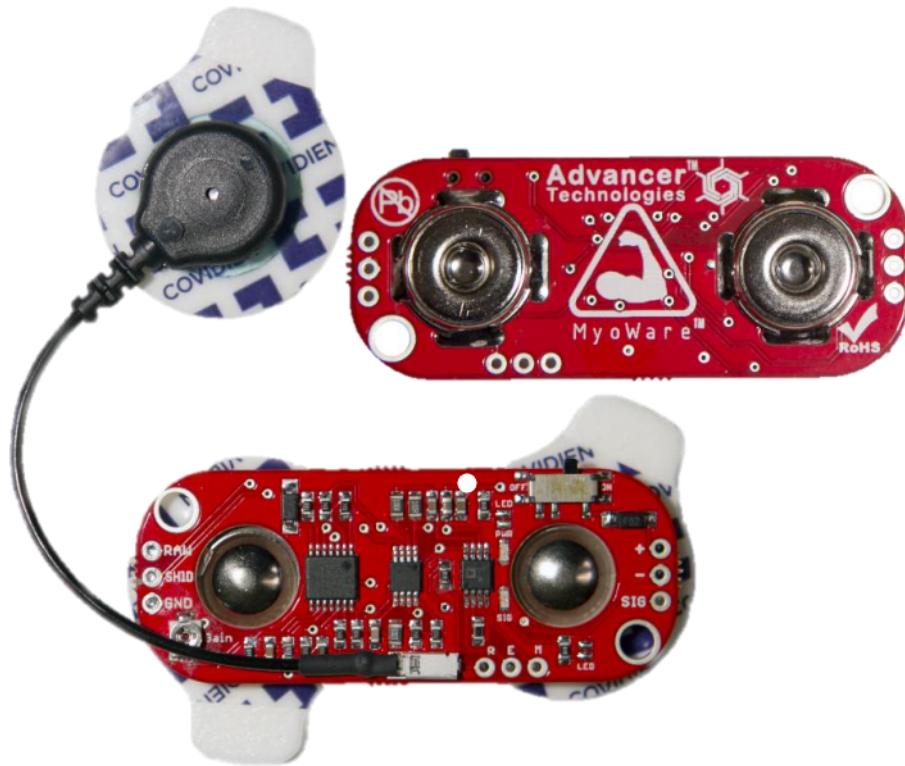


Figura 1: Sensore elettromiografico

Caratteristiche principali del sensore elettromiografico:

- Design indossabile
- Alimentazione da +3,1 V a +5,9 V
- Due modalità di uscita:
 - Inviluppo del segnale EMG
 - Misura grezza dell'EMG
- Indicatori LED
- Appositamente progettato per microcontrollori
- Guadagno regolabile

¹Datasheet MyoWare MUSCLE SENSOR (AT-04-001)

b. Power Supplies AGILENT E3631A

AGILENT E3631A^[2], riproposto in **Fig.2**, è un alimentatore ampiamente utilizzato in ambito di laboratorio per alimentare e controllare dispositivi elettronici in diverse modalità (ad esempio, tensione costante, corrente costante).

Con l'alimentatore AGILENT E3631A si è in grado di generare una tensione massimo di 25 V, una tensione minima di -25 V e una corrente di 5 A. Il dispositivo è stato progettato per essere controllato da remoto mediante interfaccia GPIB o RS232.

Nel dettaglio lo strumento può restituire in output i seguenti valori:

- **Caratteristiche d'uscita**

- Uscita 1: da 0 a 6 V, da 0 a 5 A
- Uscita 2: da 0 a +25 V, da 0 a 1 A
- Uscita 3: da 0 a -25 V, da 0 a 1 A

- **Precisione di programmazione a 25°C ±5°C**

- Tensione: 0,05% + 20 mV, 0,05% + 20 mV, 0,1% + 5 mV
- Current: 0,15% + 4 mA, 0,15% + 4 mA, 0,2% + 10 mA



Figura 2: Alimentatore

²Power Supplies AGILENT E3631A

c. Scheda acquisizione DAQ NI USB-6001

DAQ NI USB-6001 [3] (**Fig.3**) è una scheda di acquisizione dati che converte i segnali analogici, provenienti da sensori o dispositivi elettronici, in segnali digitali affinchè possano essere elaborati e analizzati da un computer.

Specifiche:

- Frequenza di campionamento: 20 kSPS
- Risoluzione: 14 bit
- Canali in ingresso: 8 (Single-ended)/ 4 (Differenziale)
- Canali in uscita: 2



Figura 3: DAQ

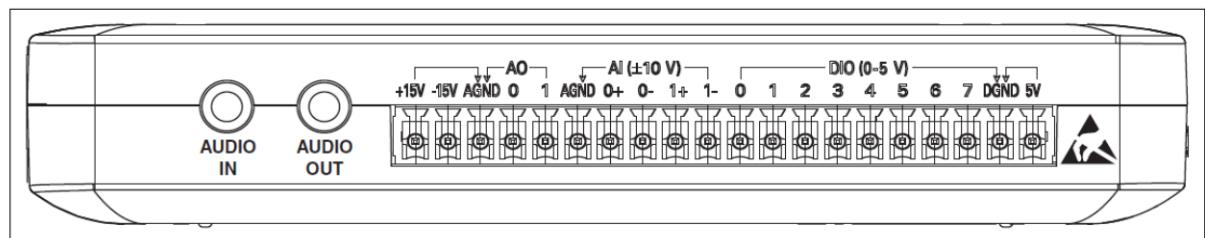


Figura 4: Morsettiera

³Datasheet DAQ NI USB-6001

d. Interfaccia di comunicazione GPIB 488

L'interfaccia di comunicazione GPIB 488^[4] (**Fig.5**) è un'interfaccia standardizzata utilizzata per connettere e controllare da remoto lo scambio di dati tra strumenti elettronici e un computer.



Figura 5: Interfaccia di comunicazione GPIB 488

e. LABVIEW 2023 STUDENT EDITION

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) è un ambiente di programmazione grafica utilizzato per:

1. la progettazione di interfacce utente;
2. il controllo di dispositivi di misura;
3. l'analisi dei dati acquisiti in tempo reale.

f. NI-MAX

NI-MAX (Measurement & Automation Explorer) è un software di gestione degli strumenti di National Instruments che consente la configurazione, il monitoraggio e il controllo di vari dispositivi di misura e controllo.

g. PERSONAL COMPUTERS

⁴Datasheet GPIB 488

3 Client: manuale utente

Il sistema di misura progettato mette a disposizione dell'utente un'interfaccia che consente di:

- Eseguire l'accesso al server mediante autorizzazione ;
- Definire il tempo di osservazione e la frequenza di campionamento per l'esecuzione della misura;
- Avviare la sessione di misura ;
- Ricevere segnalazioni dal server sia quando la misura è in esecuzione e sia quando essa è terminata ;
- Visualizzare su un grafico il segnale acquisito con i corrispettivi valore massimi e efficaci calcolati in fase di elaborazione ;
- Fornire un warning se i valori rilevati nel punto precedente sono inferiori a una certa soglia impostata dall'utente.

Nei paragrafi successivi verranno analizzati nel dettaglio, fornendo una descrizione più accurata dei parametri che l'utente deve inserire, i punti sopraelencati.

3.1 Accesso al server

L'utilizzo del sistema realizzato da parte di un utente può avvenire solo dopo che lo stesso utente ha inserito delle credenziali di autenticazione.



Figura 6: Interfaccia di accesso al Server tramite password

In **Fig.6** è riportata l'interfaccia di login dove è necessario impostare:

Password

Questo controllore consente di inserire la password (sequenza alfanumerica di lunghezza indefinita) per autorizzare l'accesso del Client. Se la password è errata o non viene inserita il sistema consentirà all'utente di inserire nuovamente le credenziali.

Address

In questo riquadro deve essere inserito l'indirizzo IP del Server con il quale bisogno instaurare la connessione.

Remote port

In corrispondenza di "Remote Port" inserire il numero della porta associato al dispositivo di destinazione con cui ci cerca di stabilire la connessione.

Una volta compilati tutti i campi (Password, Address e Remote Port) premere **AVANTI**.

3.2 Definizione dei parametri per l'acquisizione

In questo settore è possibile settare alcuni parametri che verranno inviati al Server per l'acquisizione del segnale elettromiografico (EMG):

Tempo di osservazione

Il Tempo di Osservazione, che dipende dall'analisi richiesta, può variare da pochi secondi a diversi minuti o più. Tuttavia, per molte analisi standard, un tempo di osservazione di alcuni secondi (ad esempio, da 6 a 8 secondi) può fornire dati sufficienti per comprendere l'attività muscolare durante determinati movimenti.

Frequenza di campionamento

La Frequenza di Campionamento di un segnale EMG, da definire in questo controllo, si situa nell'ordine dei kiloHertz (kHz), generalmente intorno ai 1000 Hz (1 kHz). Questa frequenza consente di catturare dettagli sufficienti delle variazioni nell'attività muscolare.

Inseriti tutti i parametri sopracitati, premere **AVANTI** se si desidera proseguire con l'acquisizione dell'EMG. In caso contrario, premere il tasto **ESCI** per bloccare la comunicazione con il Server.

SET-UP DI MISURA

Impostare i parametri sotto indicati per eseguire l'acquizione del segnale:

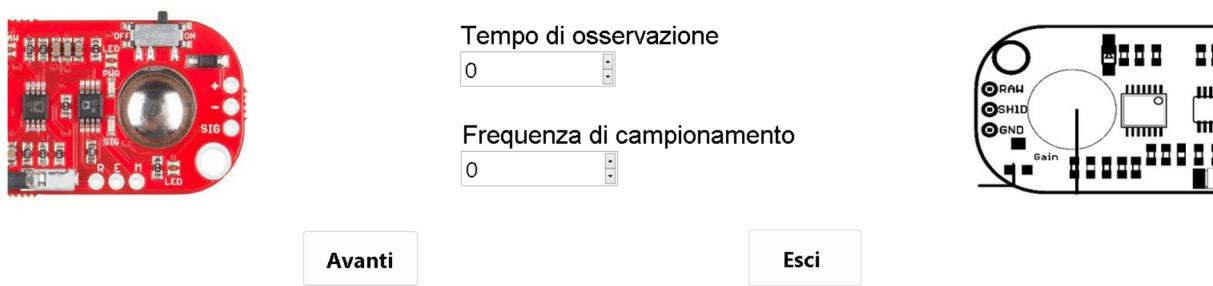


Figura 7: SET-UP di Misura

3.3 Avvio del processo di misura

Nella seguente sezione l'utente ha la possibilità di scegliere tra due opzioni:

Avvia la misura

Spuntando tale casella, il Client invia un messaggio al Server per avviare l'acquisizione del-

l'EMG. Per avere conferma che la misura è stata eseguita correttamente, l'utente riceve due segnalazioni a dimostrazione del fatto che il processo di misura è stato avviato e terminato.

Arresta il processo

Questa seconda opzione invia un messaggio al Server a valle del quale la connessione Client-Server viene interrotta.

Scegliere una delle seguenti opzioni:



Figura 8: Avvia Arresta

Selezionata l'impostazione desiderata, premere **OK**.

3.4 Impostazione parametri di soglia

In questa sezione, si consiglia di impostare dei valori di soglia minimi per il picco massimo rilevabile e per il valore efficace del segnale EMG. Si noti che se i valori inseriti scendono al di sotto della soglia dei segnali EMG, si verificherà un avviso (Warning).

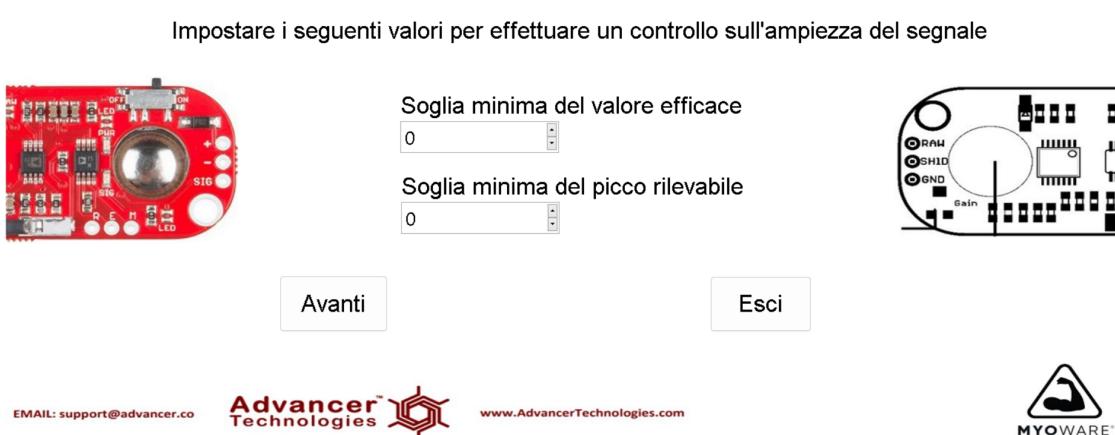


Figura 9: Valori di soglia

3.5 Visualizzazione Elettromiogramma

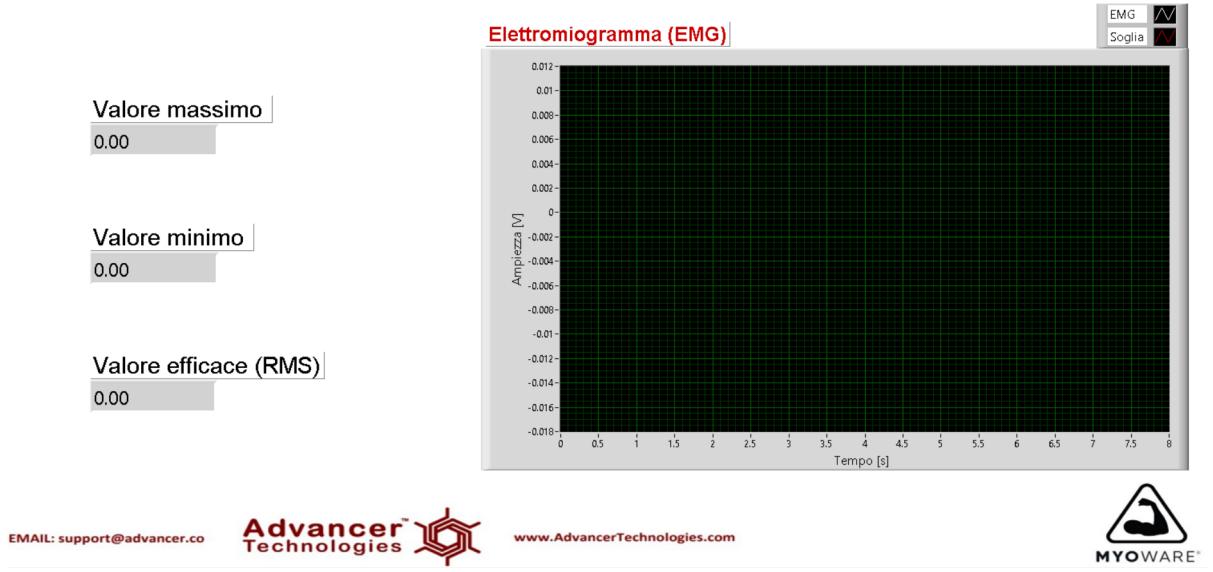


Figura 10: Elettromiogramma

4 Manuale Programmatore

Lo scopo di questa sezione è quello di descrivere e analizzare le scelte progettuali adottate per la realizzazione del programma. L’ambiente di programmazione utilizzato permette di sviluppare un file in formato VI costituito da un *Block diagram* ed un *Front Panel*. Quest’ultimo costituisce il sistema di interfaccia per l’utente dell’applicazione e contiene: i *controlli*, cioè l’input, e gli *indicatori* cioè l’output. Il primo, invece, rappresenta il codice sorgente del software in linguaggio G e presenta: controlli e indicatori del Front Panel che appaiono come terminali, nodi e funzioni.

4.1 Sub-VI

Al fine di semplificare la visualizzazione del block-diagram sono stati realizzati i seguenti *Sub-VI*:

SubVi-Pilotaggio alimentazione

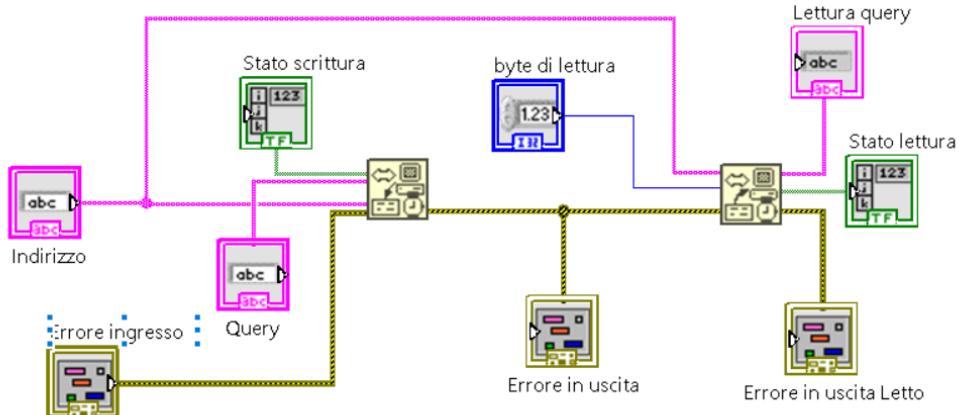


Figura 11: Pilotaggio alimentatore

Il Sub-VI sopra riportato stabilisce la connessione tra il server e l'alimentatore. In particolare, si individua una funzione *GPIB-Write* che prende in *input* due controlli: l'indirizzo dello strumento che si intende pilotare e una *query*. Quest'ultima permette di settare da remoto i parametri di alimentazione dello strumento (in particolare i valori di tensione e corrente) attraverso l'espressione *APPLY P25V,3.3,1.0*; e di interrogarlo mediante il comando *APPLY?*. A valle della funzione descritta si inserisce il blocco *GPIB-Read* che prende in input l'indirizzo dell'alimentatore e i *byte* da leggere in risposta alla *query*. In output si ottiene una stringa contenente la risposta alla *query*.

SubVI-Lettura stringa

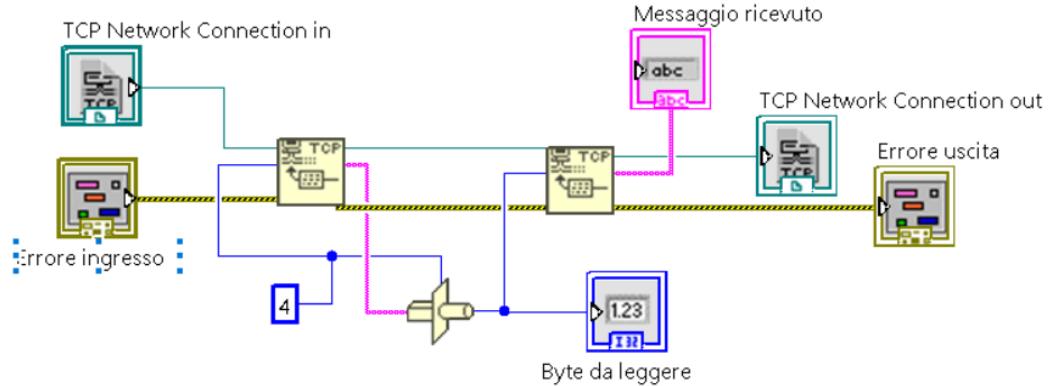


Figura 12: Lettura Stringa

Il Sub-VI realizzato per la lettura di una stringa è costituito da due funzioni che si chiamano *TCP Read*. Ciascun blocco prende in ingresso l'*errore*, il *connected ID* e restituisce in uscita la lettura di un messaggio.

In particolare, la prima operazione di lettura individua la lunghezza del messaggio, la quale, a sua volta, è convertita in un intero che viene inserito in ingresso al secondo blocco. E' bene osservare che in ingresso alla prima funzione di lettura il numero di *byte* da leggere viene fissato attraverso una costante pari a quattro: tale scelta è legata all'esigenza di dover leggere un messaggio di tipo stringa. In seguito, il secondo *TCP Read*, preso in ingresso il numero di *byte* necessari per leggere la stringa , restituisce la lettura del messaggio ricevuto.

SubVI-Scrittura stringa Server

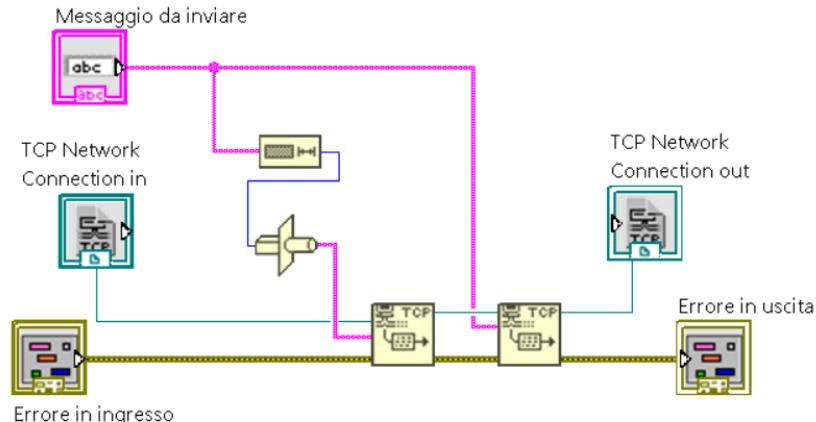


Figura 13: Scrittura Stringa Server

Il Sub-VI Scrittura stringa prevede l'impiego di due blocchi *TCP Write* che prendono in ingresso l'*errore*, il *connected ID* e il messaggio che si intende inviare. La prima funzione invia informazioni relative alla lunghezza della stringa, le quali sono ottenute attraverso l'inserimento di una *string length*. Nel secondo processo si esegue la scrittura del contenuto del messaggio.

SubVI-Lettura array

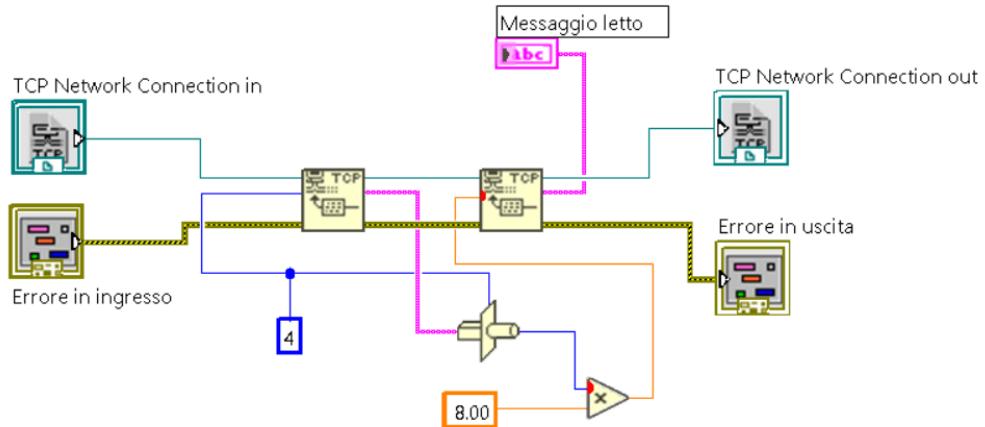


Figura 14: Lettura array

Il SubVI di lettura di un *array* differisce da quello realizzato per una stringa e per il numero di *byte* necessari alla lettura di una variabile numerica. In relazione a ciò, è bene evidenziare che lo spazio occupato da una sequenza numerica di tipo *dbl* è calcolato dal prodotto tra la lunghezza dell'array e 8, ovvero lo spazio necessario per rappresentare un singolo valore numerico.

SubVI-Scrittura array 1D

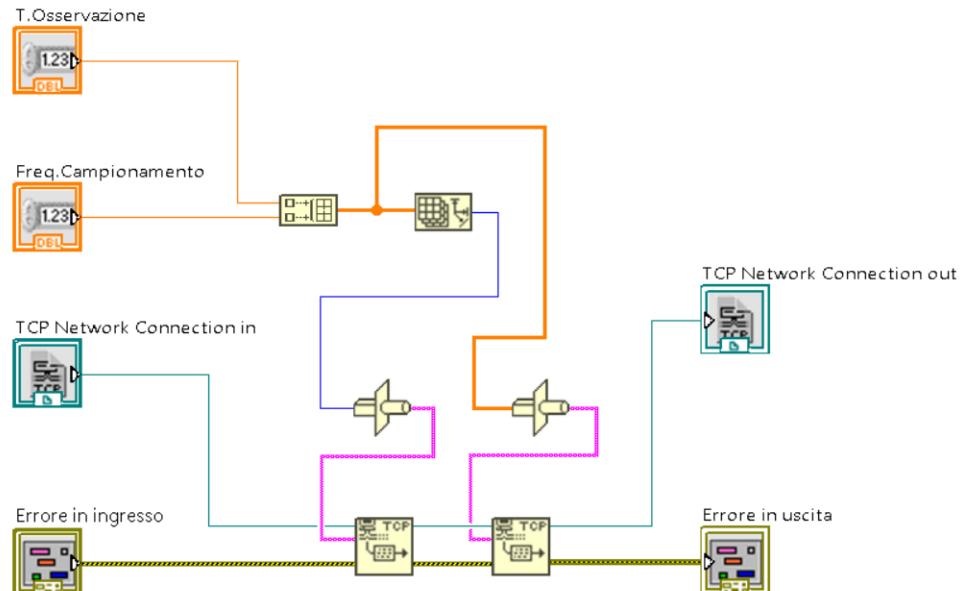


Figura 15: Scrittura Array 1D

La realizzazione del Sub-VI sopra riportato è stata necessaria per poter trasmettere informazioni di tipo diverso da quello stringa. Infatti, una delle caratteristiche della funzione *TCP Write* è quella di prendere in ingresso esclusivamente delle stringhe. A tal proposito, al fine di trasmettere contemporaneamente due messaggi di tipo *dbl*, è stato indispensabile creare un *array* attraverso il blocco *build array* e stimarne le dimensioni con l'apposita funzione (*size array*).

Tali informazioni sono state poste in ingresso ai due *TCP Write* analogamente a quanto descritto per il Sub-VI- Scrittura stringa.

SubVI-Scrittura Array 2D

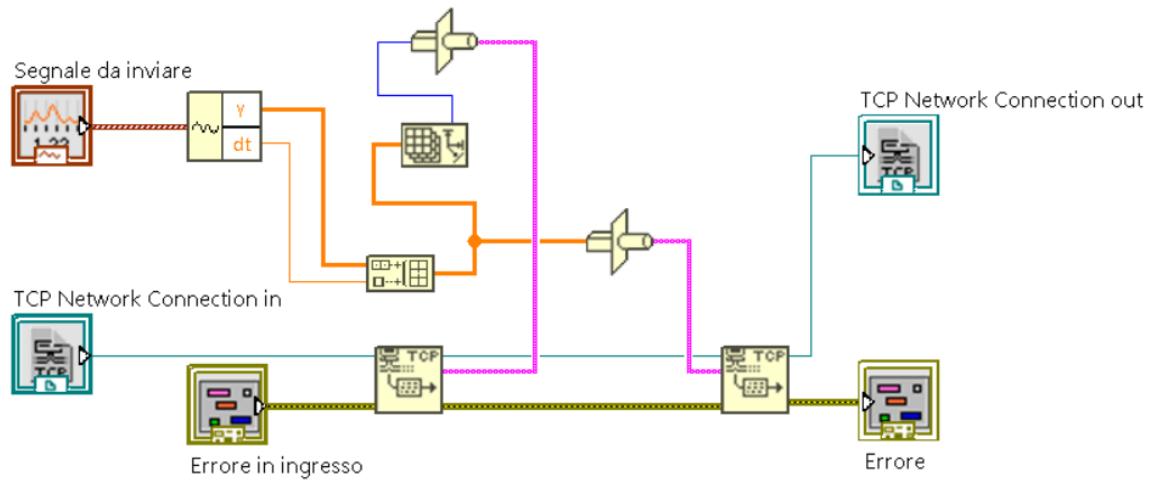


Figura 16: Scrittura array 2D

Il Sub-VI sopra riportato è stato progettato per garantire la scrittura di un array bidimensionale che, nello specifico, rappresenta il segnale elettromiografico acquisito. In particolare, l'indicatore in uscita al primo blocco di scrittura necessita di un'operazione di conversione da *Waveform* ad *array* attraverso la combinazione della funzione *Get Waveform Components* con il blocco *Build array*.

4.2 Client: manuale programmatore

Nella seguente sezione viene descritto il block diagram relativo al lato Client. Al fine di coordinare l'esecuzione delle funzioni è stato necessario impiegare una struttura di tipo *Sequence*.

Nel primo frame, illustrato nella **Fig.17**, viene modellato il processo di connessione autorizzata del client al server. L'utente ha la possibilità di inserire la password attraverso la funzione *prompt user*. Nel caso in cui la password risulti errata, è consentito all'utente di reinserire le credenziali utilizzando la struttura di controllo *while*. L'uscita dal ciclo *while* avviene in due scenari distinti: nel primo caso, la password inserita coincide con il controllo di una stringa costante predefinita; nel secondo, l'operatore booleano *avanti* viene negato.

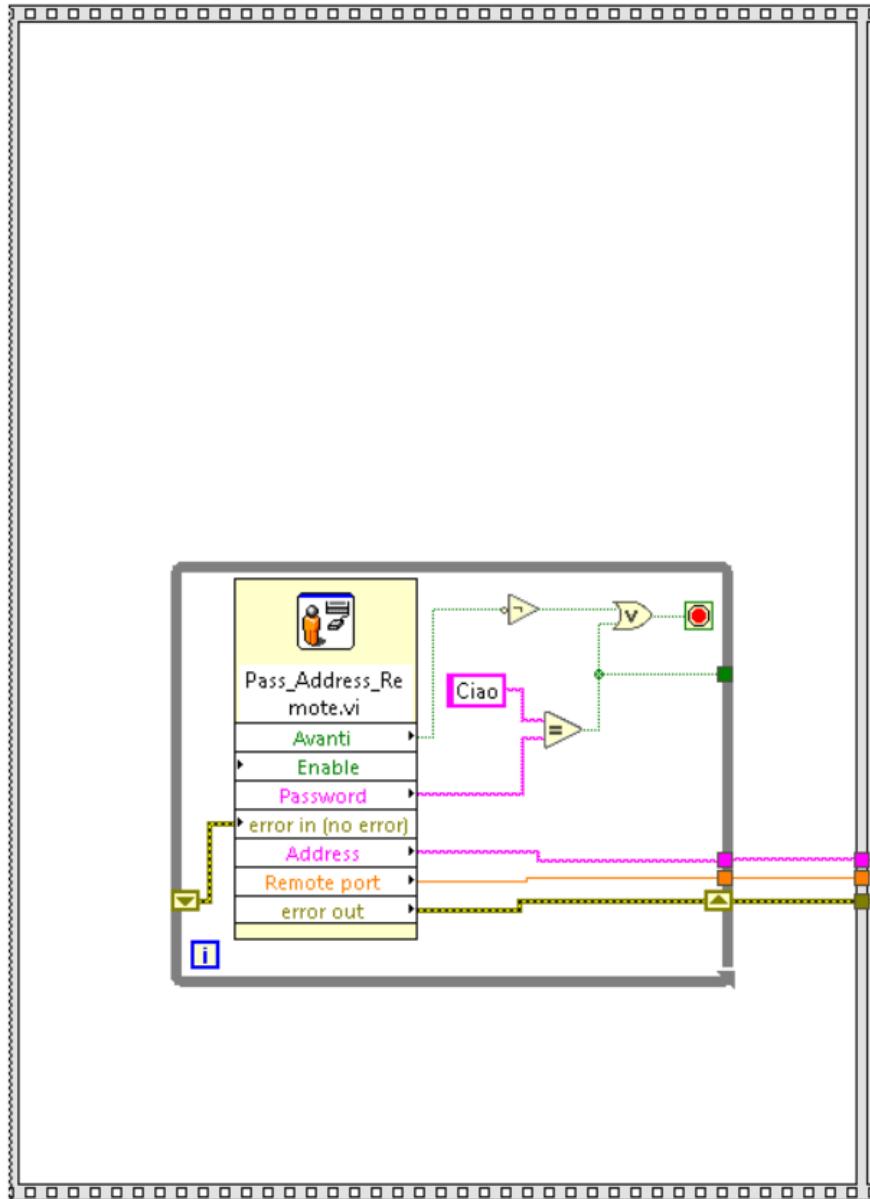


Figura 17: Frame 1 : Accesso tramite password.

Autorizzata la connessione, come riportato in **Fig.18**, l'utente ha la possibilità di inserire i parametri per acquisire il segnale attraverso la funzione *prompt user*. Tali parametri vengono

successivamente inviati al server utilizzando il Sub-VI di scrittura array descritto nei paragrafi precedenti.

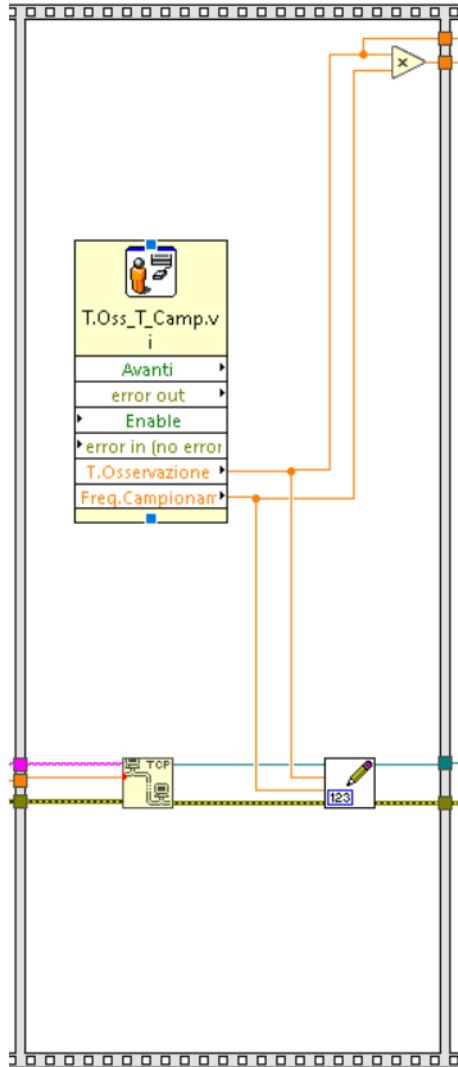


Figura 18: Frame 2 : Configurazione dei parametri di tempo di osservazione e frequenza di campionamento.

Una volta configurato il set-up di misura, attraverso il blocco *display message* l'utente ha la possibilità di effettuare due scelte: "Avvia misura" e "Arresta il processo". La selezione dell'utente è inserita come input in una struttura di tipo case, il caso false corrisponde ad *Arresta il processo*, come mostrato in **Fig.19**. In tali condizioni, il client invia al server la stringa "Arresta la misura" e successivamente riceve una notifica che conferma l'arresto, procedendo infine a chiudere la connessione con il client.

Nel caso true (**Fig.20**), corrispondente ad "Avvia la misura", il Client invia al server la stringa contenente l'opzione selezionata e successivamente riceve due notifiche: "Misura in esecuzione", confermando l'avvio dell'esecuzione, e "Misura terminata", confermando la conclusione del processo di misura. Il segnale acquisito viene letto dal Client attraverso l'impiego di due blocchi di tipo *TCP Read*. In particolare, il primo restituisce le dimensioni dell'array contenente l'EMG sotto forma di stringa. Il secondo blocco di lettura, invece, consente di leggere effettivamente il contenuto del segnale elettromiografico. Quest'ultimo, viene convertito in un valore di tipo double, e mediante l'utilizzo della funzione *Build Waveform*, le informazioni riguardanti le ampiezze e il passo di campionamento, vengono trasformate in un indicatore di tipo *Waveform*.

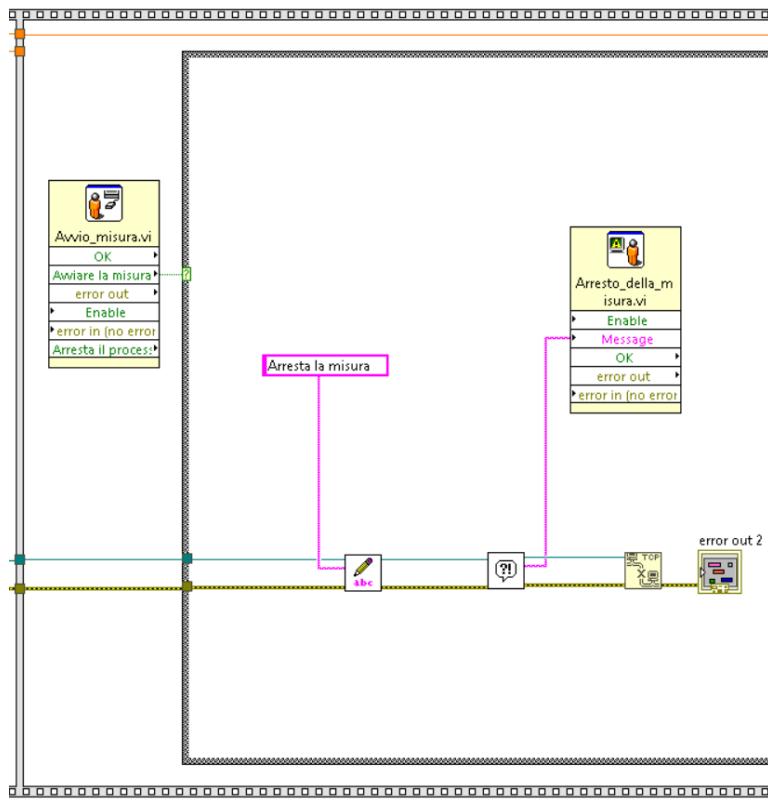


Figura 19: Frame 3 : Avvio misura (Caso false).

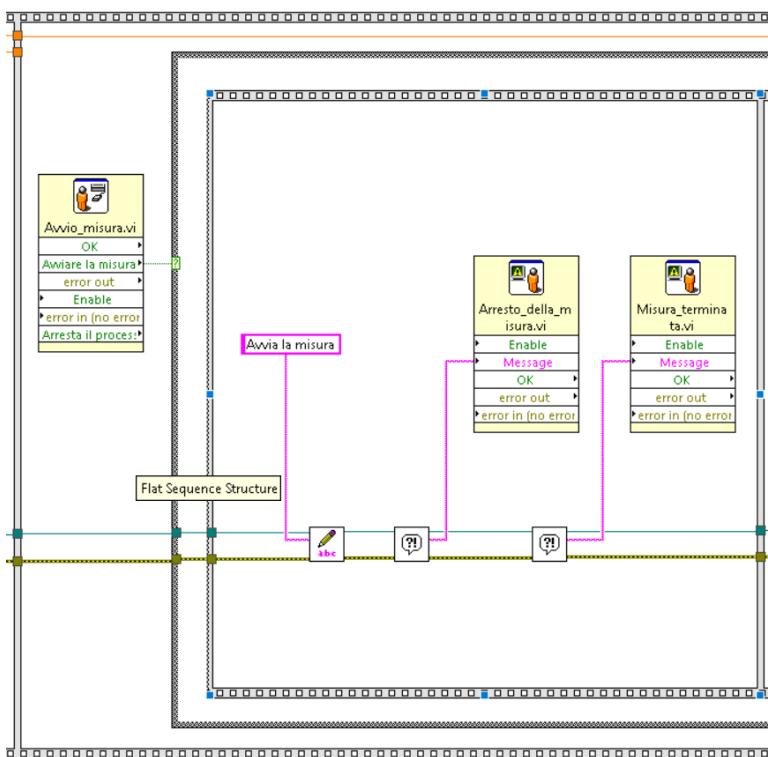


Figura 20: Frame 3 : Avvio misura (Caso true).

Successivamente, il segnale è sottoposto alle seguenti elaborazioni:

- Calcolo del valore massimo e minimo attraverso l'utilizzo della funzione *Waveform Min Max*;
- Calcolo del valore efficace mediante la funzione *Basic DC-RMS*.

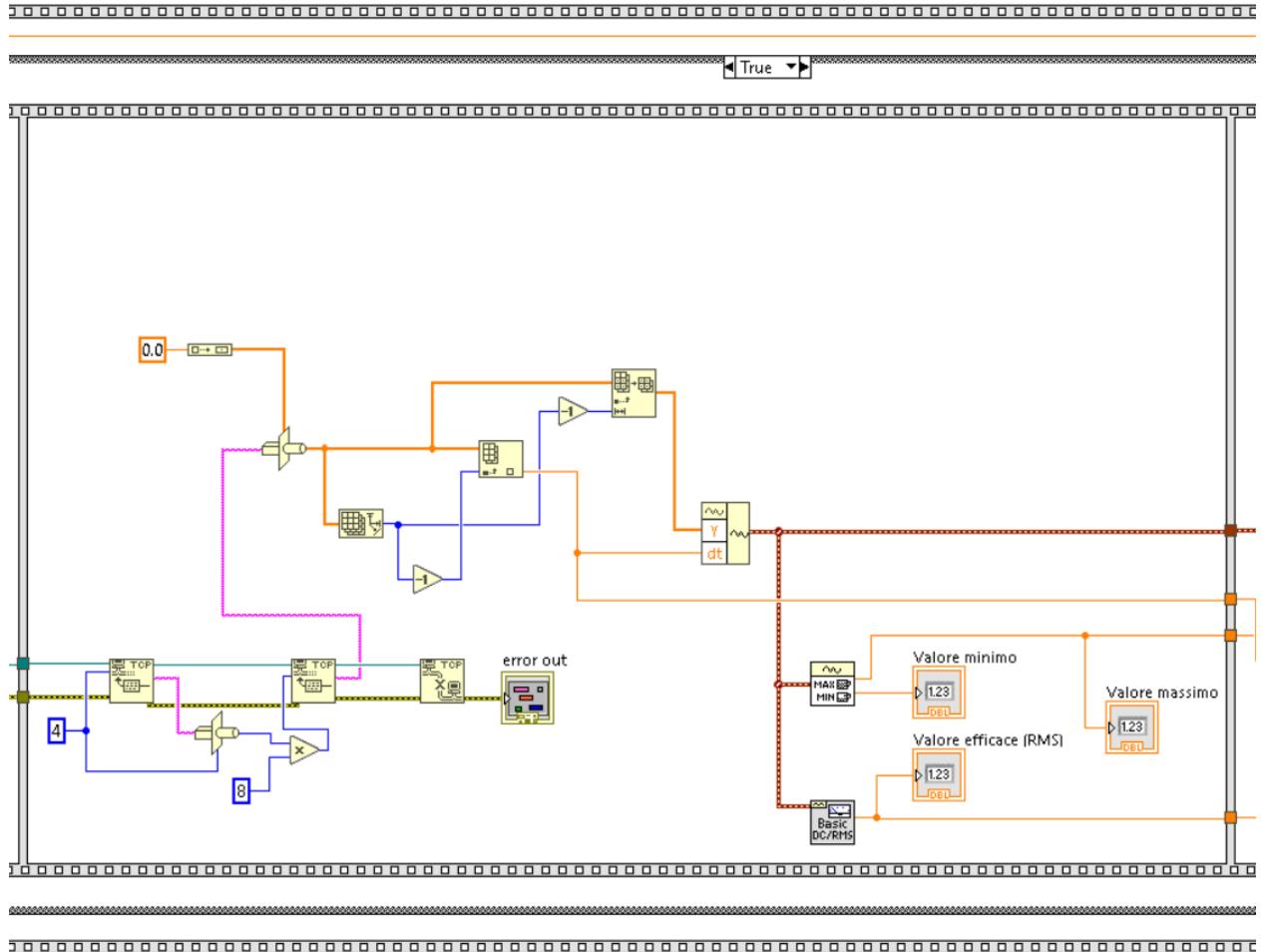


Figura 21: Frame 3 :Acquisizione ed elaborazione.

Nell'ultimo frame, l'obiettivo è fornire un avviso relativo ai valori di picco e al valore efficace del segnale. Utilizzando la funzione "Prompt User", l'utente stabilisce la soglia minima per il valore di picco e l'RMS. Successivamente, esegue un confronto con i valori calcolati sul segnale per il controllo della struttura "case". Nel caso in cui i valori risultassero inferiori alle soglie stabilite, si verifica la condizione "true", che attiva la visualizzazione di un avviso sul display mediante la funzione *Display Message*. In caso contrario, il programma procede alla rappresentazione grafica del segnale elettromiografico e delle corrispondenti soglie scelte.

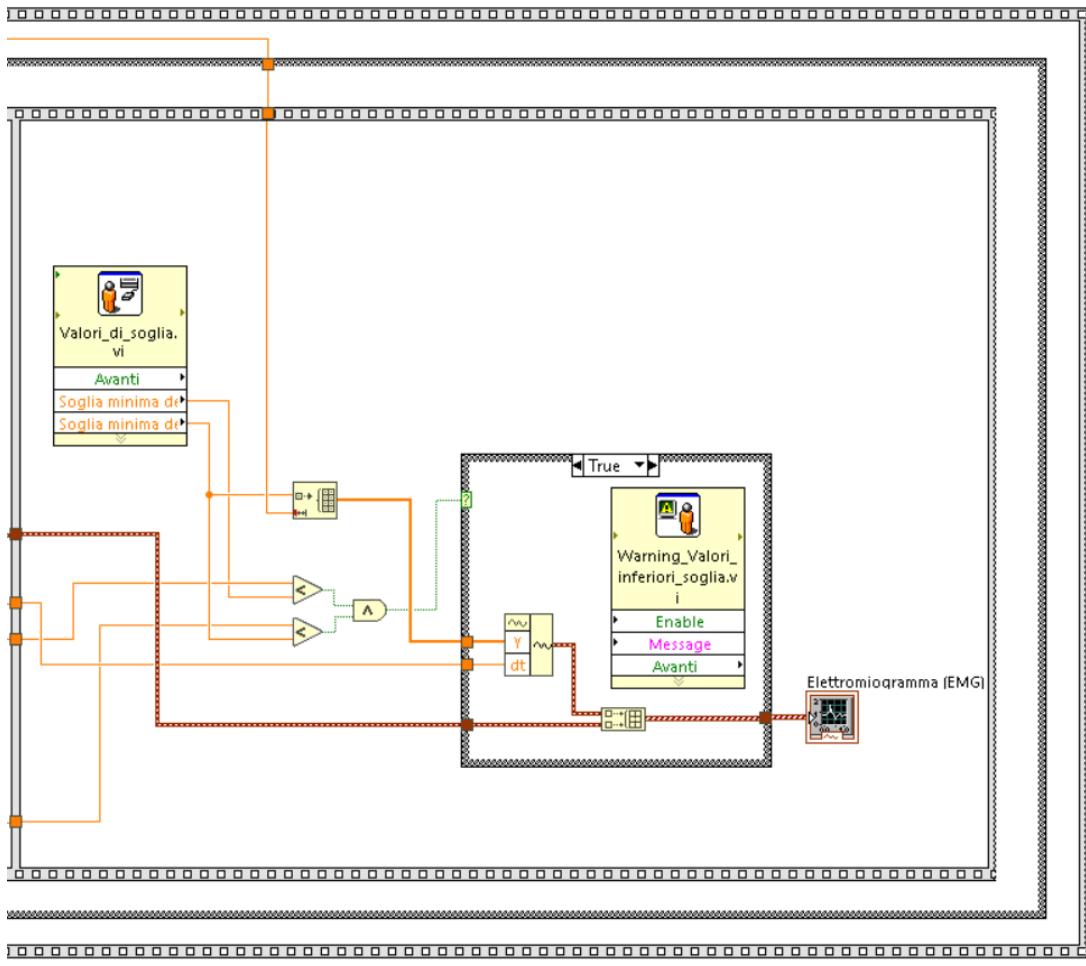


Figura 22: Frame 4 : Scelta della soglia minima di picco rilevabile e valore efficace.

4.3 Server: manuale programmatore

In questa sezione, si fornisce una dettagliata descrizione del block-diagram relativo al lato server. L'intero schema è basato su una struttura di tipo *Sequence*, progettata per coordinare l'ordine di esecuzione delle diverse funzioni.

Successivamente, verranno esaminati in modo approfondito i quattro frame che compongono la struttura del lato server.

Il primo frame, riportato in **Fig.23**, definisce il meccanismo di pilotaggio dell'alimentatore attraverso l'interfaccia di comunicazione GPIB IEE-488.

Il secondo frame (**Fig.24**) implementa e stabilisce la connessione con il client tramite protocollo TCP-IP. In dettaglio, si impiega il blocco *TCP-IP listen*, il quale include un controllo in cui è possibile specificare il numero della porta di comunicazione. Quest'ultimo deve essere specificato dal lato client per stabilire il trasferimento dei dati. Dopo aver stabilito la connessione, il server procede poi alla lettura dei valori di tempo di osservazione e frequenza di campionamento. Tali parametri, inviati in formato stringa, vengono convertiti nel formato *dbl* per essere utilizzati in fase di acquisizione. L'array fornito dal client, mediante l'impiego della funzione *index array*, permette di estrarre il tempo di osservazione in posizione 0 e la frequenza di campionamento in posizione 1. Il loro prodotto fornisce il numero di campioni necessari alla conversione del segnale.

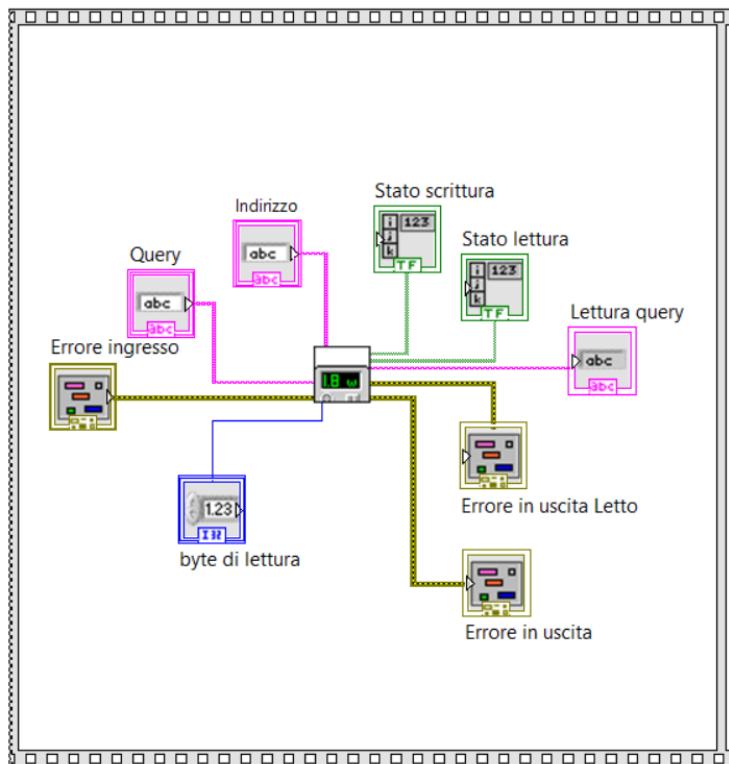


Figura 23: Frame 1 : Pilotaggio dell'alimentatore

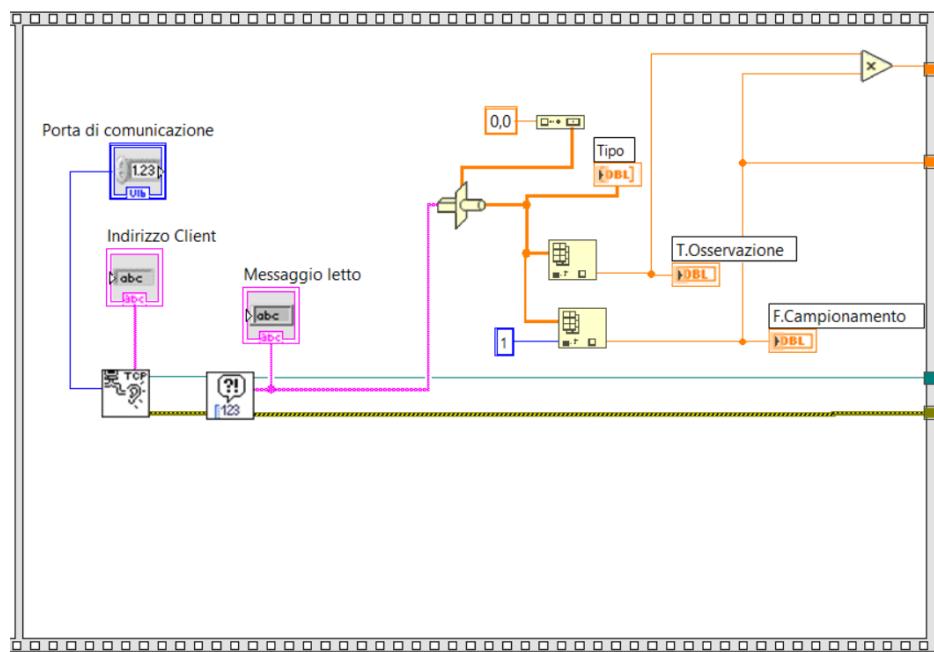


Figura 24: Frame 2 : Comunicazione con il Client

Il terzo frame (**Fig.25**) è progettato per avviare o arrestare il processo di misura. Questo meccanismo è modellato grazie all'utilizzo della struttura *case*, la quale necessita di un controllo booleano che permette dallo stato logico true allo stato logico false. Lo stato logico true viene raggiunto quando la stringa letta dal server coincide con la costante "Avvia la misura". In tal caso, il server

invia un messaggio al client in ascolto per confermare l'esecuzione della misura. In questo modo, viene avviata la procedura di misurazione tramite l'utilizzo di tre funzioni:

1. Create Channel (AI-Voltage);
2. Timing (Sample Clock);
3. Read (Analog Wfm 1channel Nsamples).

Il blocco *create channel (AI-Voltage)* permette di definire il canale fisico da utilizzare prendendo in input i seguenti controlli:i canali di ingresso analogici e i valori di fondo scala minimo e massimo. L'output, della funzione descritta, stabilisce un riferimento per il blocco *Timing (Sample Clock)* il quale permette di gestire la parte di campionamento del canale analogico. Esso prende in ingresso: il Task out proveniente dalla funzione a monte, una serie di parametri che definiscono il set up di misura (Frequenza di campionamento e numero di campioni) e la modalità di acquisizione impostata di default su *Finite Samples*. Infine, il blocco di *Read (Analog Wfm 1channel Nsamples)* consente di leggere i dati acquisiti e riporti in un indicatore di tipo *Waveform Graph*. Conclusa la fase di acquisizione, il server invia un messaggio di "Misura terminata" al Client. In seguito,i dati del Waveform vengono trasmessi per essere elaborati dal lato Client. Successivamente viene interrotta la comunicazione.

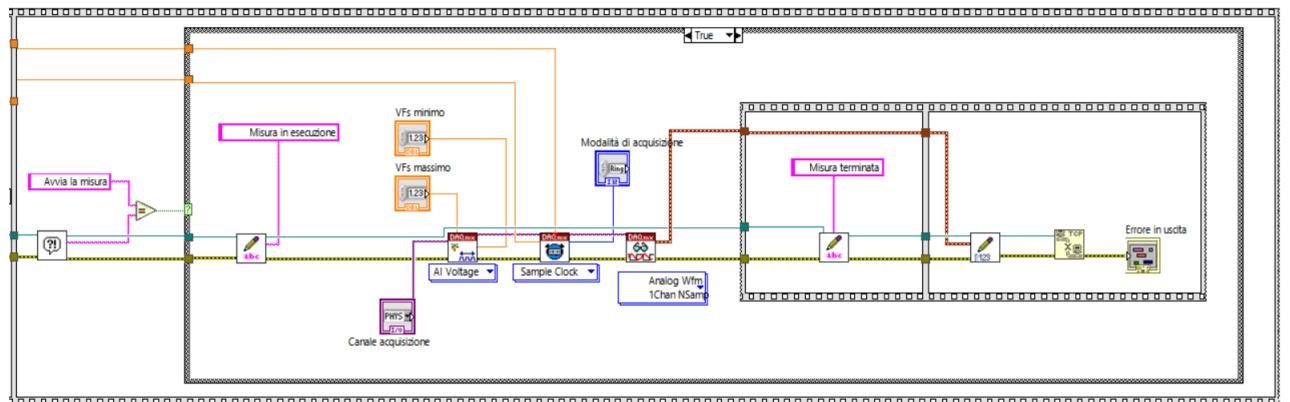


Figura 25: Frame 3 : Elaborazione del segnale EMG (Caso True)

Lo stato logico false si verifica quando la Stringa letta risulta diversa da "Avvia misura".In questo caso, il Server invia una segnalazione di errore al lato Client e termina la connessione.

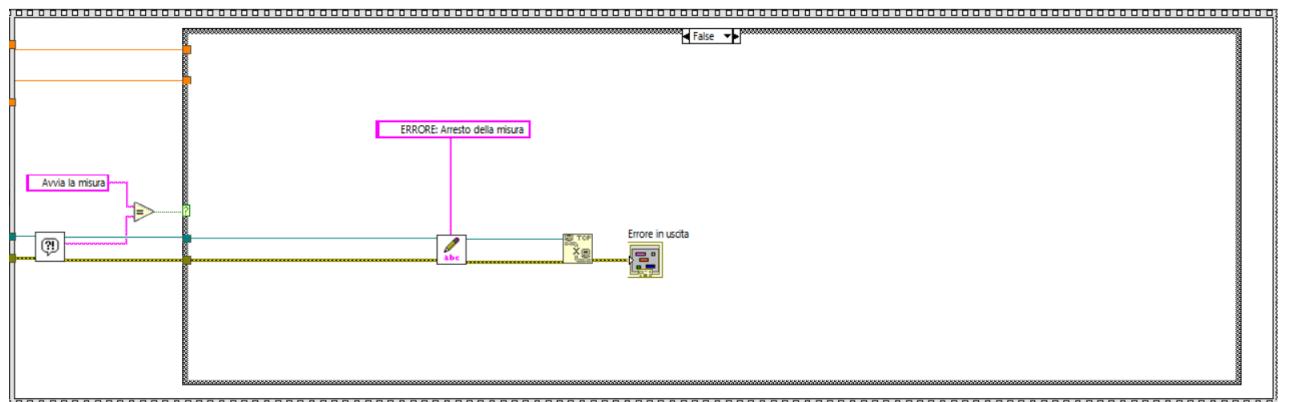


Figura 26: Frame 4 : Elaborazione del segnale EMG (Caso False)

5 Conclusioni

La soluzione integrata ha fornito dati precisi sull'attività muscolare, agevolando la comprensione dei pattern di contrazione muscolare e fornendo informazioni cruciali per applicazioni biomeccaniche, mediche o di ricerca nell'ambito della fisiologia umana e del movimento.