

ALMA MATER STUDIORUM
UNIVERSITA' DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA
Sede di Forlì

Corso di Laurea in
INGEGNERIA MECCANICA
Classe L-9

ELABORATO FINALE DI LAUREA
in Macchine

**DEFINIZIONE DEL SISTEMA DI ACQUISIZIONE E
CONTROLLO DI UNA SALA PROVA MOTORI**

CANDIDATO
Ugo Scarpellini

RELATORE
Prof. Ing. Enrico Corti

Anno Accademico 2014/2015
Sessione II

Indice

1	Abstract	5
2	Sala Prova	6
	2.1 LabView	9
	2.2 TestIt	11
3	Bilancia AVL 733s	13
	3.1 Dispositivo	13
	3.1.1 Principi Operativi	15
	3.1.2 Circuito di Misura	17
	3.2 VI AVL_733	20
	3.2.1 Segnali di Input e Output	20
	3.2.2 Interfaccia Grafica	22
	3.2.3 AK Protocol	26
	3.2.4 rMain	33
	3.2.5 SubVI	35
	3.3 Altre Applicazioni	42
4	Smoke Meter AVL 415s	43
	4.1 Dispositivo	43
	4.1 .1 Principi Operativi	45
	4.1.2 Parametri di Misura	48
	4.1.3 Fattori di Influenza	50
	4.2 VI AVL_415s	53
	4.2.1 Segnali di Input e Output	53
	4.2.2 Interfaccia Grafica	54
	4.2.3 AK Protocol	58

	4.2.4 AVL_415S	61
	4.2.5 SubVI	66
5	MDrive 23 Plus	67
	5.1 Dispositivo	67
	5.2 VI R_NEMA_Main	69
	5.2.1 I segnali di Input e Output	69
	5.2.2 Interfaccia Grafica	70
	5.2.3 MCode	72
	5.2.4 Main e subVI	74
6	Integrazione in TetIT	81
	6.1 Gestione de Dati	81
	6.1.1 Protocollo di Comunicazione	81
	6.1.2 R_gl	82
	6.1.3 FIFO	83
	6.2 Struttura dei SubVI	83
	6.2.1 AVL 733s	84
	6.2.2 AVL 415s	85
	6.2.3 NEMA32	87
	6.3 rMain	88
7	Appendice	89
	Bibliografia	92

1 – Abstract

L'obiettivo della tesi è l'integrazione di alcuni dispositivi AVL e di un motore elettrico passo-passo lineare all'interno di un sistema di controllo e acquisizione dati per una sala prova motori. La sala prove in cui è svolto il lavoro è quella del laboratorio Hangar del Dipartimento di Ingegneria Industriale di Forlì.

Inizialmente viene proposta una breve descrizione della sala prove; essa è divisa in una parte hardware, in cui vengono spiegati i principali *device* utilizzati per il controllo, e in una parte software, in cui vengono descritti i programmi utilizzati (**LabView** e **TestIT**).

Successivamente, i capitoli centrali, si occuperanno dei componenti AVL e del motore lineare utilizzati in sala.

Per quanto riguarda AVL i dispositivi da integrare sono la bilancia per il carburante **733s** e lo smoke meter **415s**. Si traccia quindi una descrizione degli impianti e si indicano i principi operativi. Entrambi sfruttano il linguaggio seriale, quindi si procede con la creazione di un'interfaccia grafica in grado di convertire i comandi desiderati dall'utente in codice **ASCII**. Allo stesso modo viene affrontato il motore lineare **NEMA**. In questo caso il dialogo avviene tramite una connessione Ethernet.

La parte centrale si conclude con un capitolo nel quale si spiega l'introduzione dei VI all'interno di TestIT e le problematiche che ne possono scaturire.

Nell'appendice finale saranno descritte brevemente le connessioni pratiche e gli interventi di progettazione avvenuti per organizzare il rack.

2 – Sala Prove

La sala prove è il laboratorio dove avvengono i test per il rilevamento delle caratteristiche meccaniche del motore quali potenza, coppia, consumi ma anche rilievi più approfonditi quali l'acquisizione della pressione nel cilindro, per analizzare nel dettaglio il processo di combustione *indicating*.

Lo strumento principale è il banco prova costituito da un basamento e da un freno che può essere meccanico, idraulico o a elettrico. Il **basamento** prende il nome di base sismica ed è la struttura alla quale si fissa il motore. Il **freno** invece si collega direttamente all'albero motore generando la coppia resistente utile a misurare le caratteristiche.

Il banco utilizzato durante l'attività di tesi è dell'azienda **Borghi & Saveri s.r.l.** a correnti parassite tipo FE 350-S con alimentazione dell'acqua di raffreddamento a circuito chiuso.

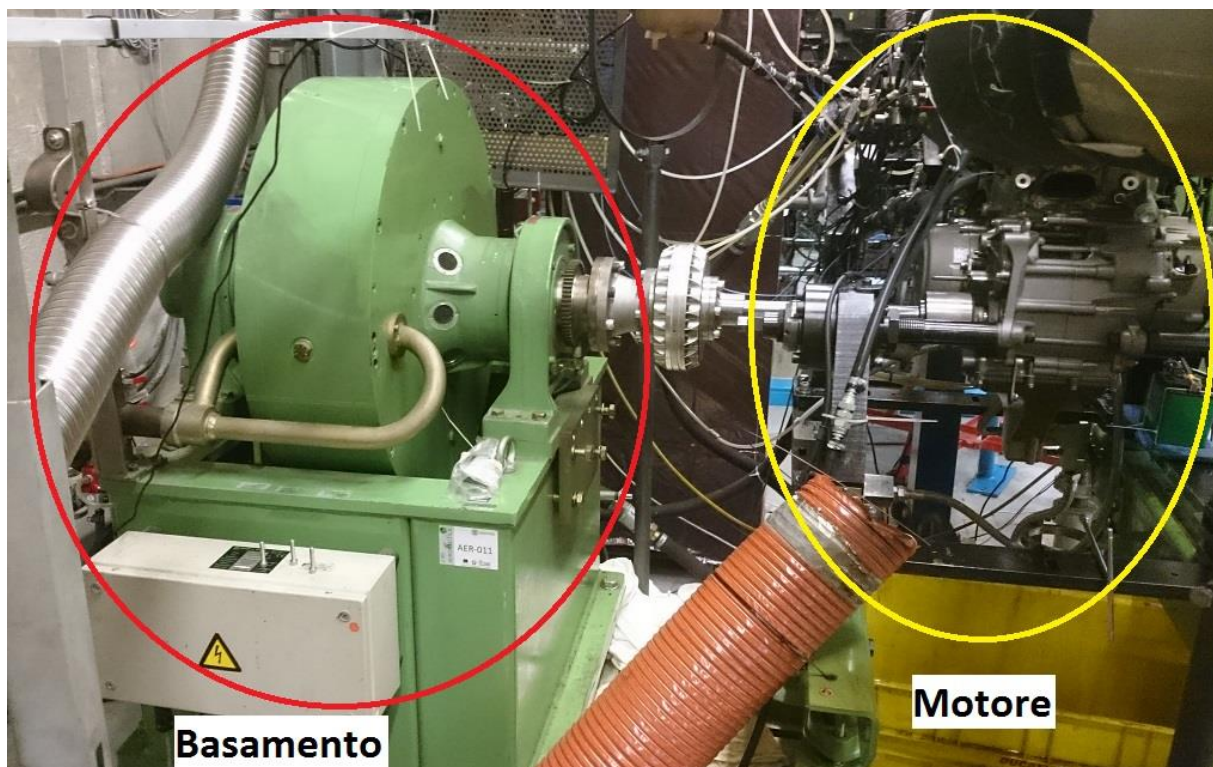


Figura 1 Banco prova Borghi & Saveri FE 350-S

Tramite un ruota dentata a 60 poli, il banco calcola i giri motore e la frequenza pickup TTL, mentre in base alle correnti parassite (che applicano il carico resistente) la coppia (raw o trattata) grazie a schede di potenza (SAT 2000) interne al proprio sistema di calcolo (DCU 2000).

Il sistema di controllo che riceve questi segnali è composto da diversi dispositivi che

costituiscono l'hardware della sala prove fornito da **National Instruments**.

Lo schema seguente mostra le connessioni fra tutti i dispositivi presenti:

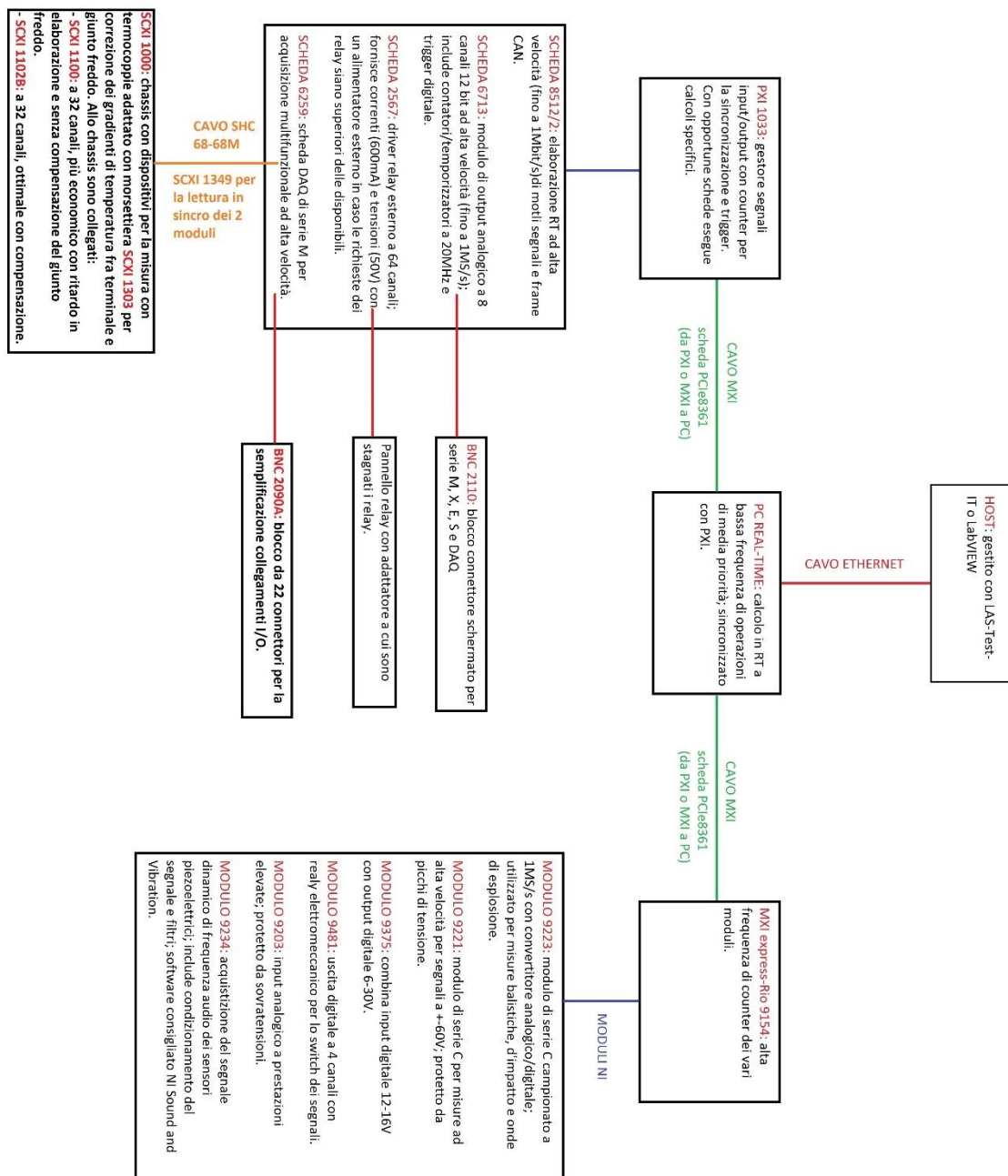


Figura 2 Schema del Layout hardware

Partendo dall'alto troviamo **HOST**, ossia il computer dell'utente. Su di esso girerà l'interfaccia grafica di **TestIT** (descritto nel capitolo successivo) nella quale si ricevono e si inviano dati alla cella.

Il pc è collegato all'interno della sala tramite un cavo Ethernet ad un ulteriore computer: è il **Real-Time** che non ha sistema operativo, ma ha la funzione di gestione a bassa frequenza

delle informazioni che si scambiano cella e HOST. Inoltre esegue anche operazioni di media priorità necessarie alla temporizzazione dei dati.

Dal PC RT escono due collegamenti differenti mediante cavo **MXI** che portano a:

- **MXI express-Rio 9154**, chassis FPGA configurabile con 8 moduli input/output della serie C;
- **PXI 1033**.

Questa configurazione permette elevate capacità computazionali (Pc RT), modularità e flessibilità attraverso i *device* collegati via MXI oltre che la completa reversibilità di moduli e schede già disponibili allo stato dell'arte.

A entrambi i dispositivi sono applicate schede NI (2567, 6259, 9375, ...) a seconda del sensore che occorre collegare al sistema di controllo. Il flusso così suddiviso rende facile la standardizzazione del sistema di controllo qualora si volesse sostituire o aggiungere un nuovo dispositivo.



Figura 3 PXI 1033 National Instrument

All'interno della cella sono presenti anche la bilancia **AVL 733s** e lo smoke meter **AVL 415s** che verranno descritti in seguito assieme alla loro integrazione nel sistema.

Infine fa parte della sala tutto ciò che riguarda la ventilazione, per il raffreddamento del motore e l'aspirazione dei gas di scarico, e le pompe che gestiscono i fluidi di raffreddamento in ingresso al motore.

2.1 – LabView

LabView (Laboratory Virtual Instrumentation Engineering) è un ambiente di sviluppo fornito da **National Instruments**. Tramite la programmazione grafica, detta Linguaggio G, permette la creazione di applicazioni di controllo, misura e analisi sfruttando i dispositivi che la NI stessa fornisce. La semplicità di un diagramma di flusso permette a questo tipo di linguaggio di essere fra i più utilizzati industrialmente.

Le applicazioni prendono il nome di VI (*virtual instruments*) e sono composte da due parti:

- **Pannello frontale:** è l'interfaccia utente; la dicitura pannello deriva dal fatto che il prodotto finale è del tutto simile ad una plancia di controllo con interruttori e potenziometri, ma anche menu e finestre tipiche dell'ambito informatico.
- **Schema a blocchi:** fisicamente è tutto quello che è presente dietro al pannello frontale; è il linguaggio di programmazione in forma grafica dove le righe di comando sono sostituite da fili contenenti dati e le funzioni sono rappresentati da blocchi.

Il pannello frontale viene realizzato con controlli ed indicatori che rappresentano i terminali del flusso di dati in ingresso e in uscita. I controlli non sono altro che gli input che l'utente fornisce al sistema e sono

rappresentati da potenziometri, pulsanti, menu a tendina e in generale tutto ciò che può rappresentare uno strumento di misura. Gli indicatori invece simulano gli output del sistema e quindi mostrano all'utente i dati acquisiti ed elaborati dal sistema; sono rappresentati da LED (per segnali logici), grafici, tabelle, matrici, termometri.

Lo schema a blocchi invece

è una diretta conseguenza del pannello: tutti i suoi input ed output diventano terminali

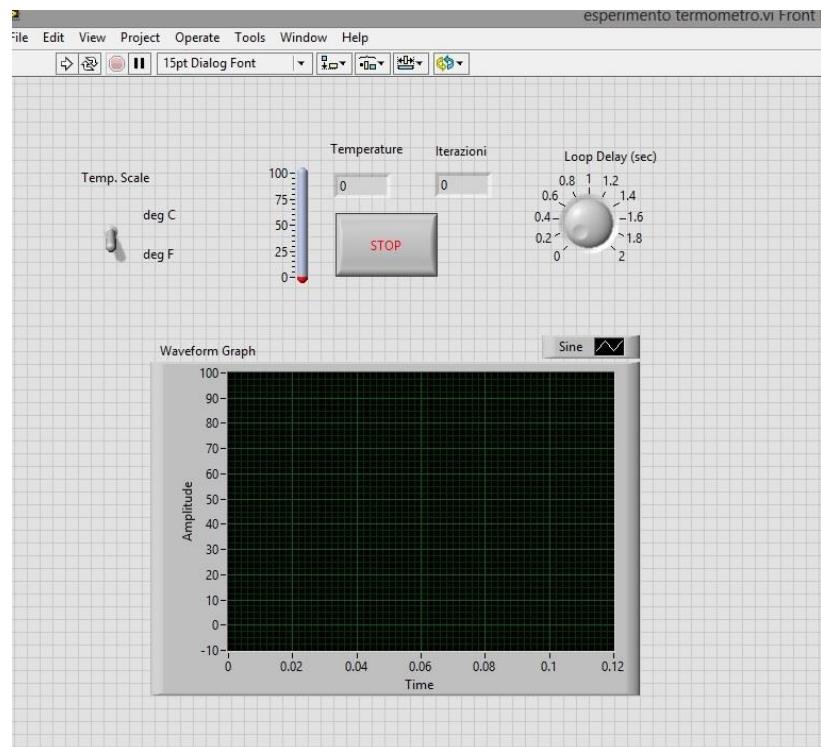


Figura 4 Esempio di pannello frontale

dell'algoritmo che vogliamo andare a realizzare. Lo schema contiene quindi il codice sorgente del nostro VI in formato grafico. Al suo interno abbiamo, oltre ai terminali, diversi elementi:

- **Funzioni:** sono rappresentate da blocchi quadrati; svolgono i compiti fondamentali richiesti dal nostro algoritmo, da semplici operazioni lineari, fino al calcolo matriciale, comprendendo la creazione di grafici e l'elaborazione di stringhe di testo.
- **Costanti:** dati numerici, lettere o segnali logici che rimangono fissi all'interno del codice.
- **Strutture:** tipiche della programmazione sono operazioni fondamentali nella creazione di VI; comprendono cicli FOR o WHILE, ma anche registri di scorrimento, *case structure* (che agiscono diversamente a seconda dell'input) e *timed loop* utili alle operazioni Real-Time.
- **Wire:** sono i fili che portano i dati da un elemento sovrapposto all'altro; a seconda del contenuto (numerico, double, booleano, stringa) assumono un colore differente, mentre la forma deriva dal tipo (scalare, array 1D, array 2D o cluster) di dato.

Di seguito è riportato lo schema a blocchi del pannello frontale precedente:

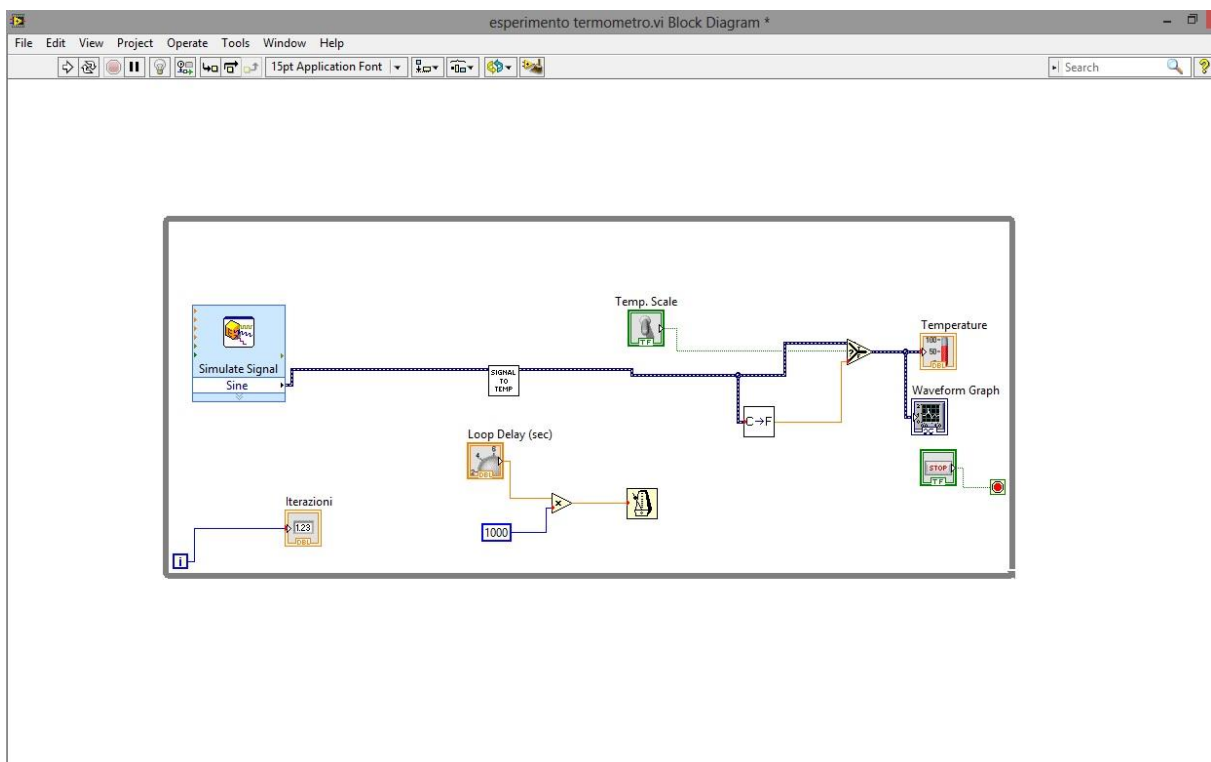


Figura 5 Schema a blocchi relativo al pannello dell'immagine 4

Poniamo infine l'attenzione su due aspetti che saranno fondamentali nel lavoro della tesi: i **SubVI** e l'elaborazione delle **stringhe**.

I SubVI sono sottoprogrammi del tutti simili ai VI che vengono richiamati sia per evitare che

il codice diventi ridondante sia per semplificare gli schemi. Essendo utilizzati come nodi dello schema a blocchi richiedono una caratteristica fondamentale: il **riquadro dei connettori**.

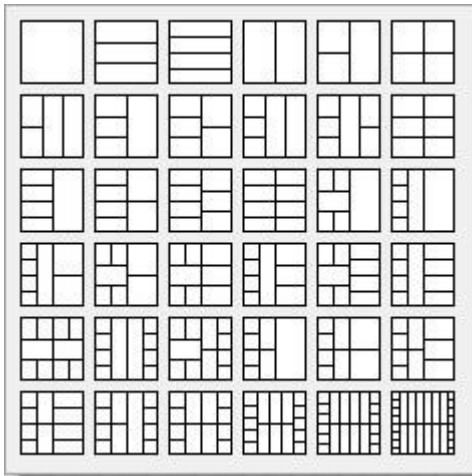


Figura 6 Pattern predefiniti da LabView

Tale espediente rende possibile il collegamento all'interno dello schema andando a definire gli input ammessi e gli output forniti dal SubVI, come se fosse una funzione standard del programma stesso. Labview permette scegliere alcuni pattern predefiniti formati da caselle a cui vengono assegnati gli ingressi e le uscite. L'elaborazione delle stringhe sarà invece fondamentale per dialogare con i dispositivi AVL che utilizzano un linguaggio seriale ASCII. Grazie a LabView abbiamo funzioni che permettono di creare messaggi, convertire

stringhe in numeri e viceversa ma abbiamo anche la possibilità di concatenare, calcolare le lunghezze, cercare all'interno dei messaggi o tagliarli a nostro piacimento.

2.2 – TestIT

Tutto l'hardware della sala è controllato da TestIT, software sviluppato da **Alma Automotive** in LabView per l'analisi e la gestione di un qualunque sistema dinamico (attuatori, sala prove, dispositivi AVL, ...). Il software si basa su un progetto che agisce in Real Time grazie al pc RT presente in sala. I VI principali sono **H_Main** e **rMain**.

H_Main è l'interfaccia che agisce sul pc HOST; i comandi forniti dall'utente e i dati in arrivo dalla sala vengono scambiati tramite variabili globali inviate al RT tramite connessione TCP e lette attraverso una connessione UDP.

rMain è invece il responsabile di tutta la gestione della sala; agisce sul pc RT presente in cella e coordina sia i dati in uscita provenienti dai device NI, sia quelli in arrivo dall'utente.

Bilancia e smoke meter vengono integrati in rMain per eseguire i calcoli, mentre i risultati, una volta portati all'HOST, saranno disponibili nei menu dell'interfaccia già presente.

Grazie a TestIT sono possibili differenti modalità di misura:

- **Manual Mode**: è la modalità di "default" in cui si avvia il software; la sala prove viene gestita con comandi da tastiera o console sui quali agisce direttamente l'operatore.

- **Semi-Automatic Mode:** permette di selezionare i setpoint del sistema di controllo muovendosi all'interno di una mappa precedentemente compilata.
- **Step Mode:** modalità automatica tipo "Statechart": è possibile scrivere una successione di condizioni di funzionamento per ciascuna delle quali è possibile monitorare una grandezza: in funzione di questa si stabilisce con quale dei breakpoints successivi proseguire.
- **Automatic Mode:** modalità automatica in cui si impone alle variabili controllate di seguire un profilo in funzione del tempo.

Tramite i menu setup è possibile indicare gli input e gli output (con le relative unità di misura) che si vogliono visualizzare a schermo a seconda della strumentazione implementata, eventualmente modificandone la linearizzazione del segnale.

Inoltre grazie ad un menu di Virtual Channel è possibile generare combinazioni di Input, che vengono calcolate direttamente in HOST, in base ai dati che arrivano dalla cella.

Ogni prova viene acquisita e registrata dal sistema con i dati relativi alla sala (banco prova, dati del motore, ...) in modo da poter essere consultata anche a posteriori per fare ulteriori analisi. La parte relativa al calcolo *Indicating* viene invece eseguita da **OBI**, un altro hardware Alma Automotive che comunica con TestIT, dando la possibilità di sincronizzare i dati.

Infine, TestIT mette a disposizione diversi controllori PID per controllare attuatori utilizzati nel banco prova (le costanti di guadagno sono programmabili dall'utente rispetto ai dati di default) e un sistema di allarmi collegato ad ogni dispositivo della sala che richiede un controllo per eseguire prove in sicurezza (banco, pompe, ventilatori, alimentazione, ...).

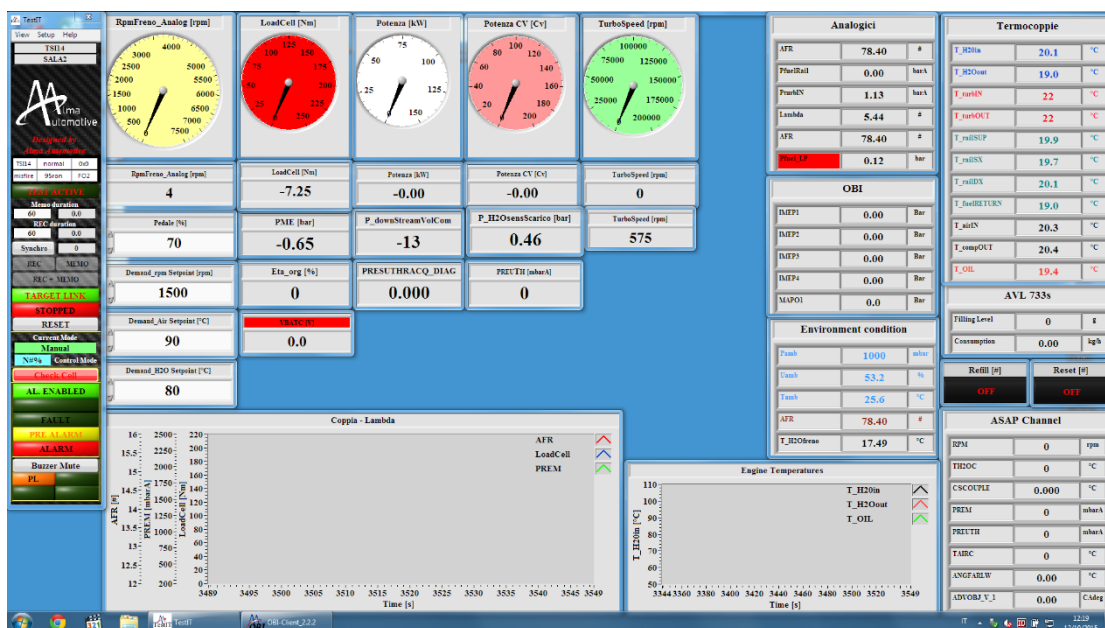


Figura 7 TestIT in esecuzione su pc Host

3 - Bilancia AVL 733S

3.1 Dispositivo

La **Bilancia AVL 733S** è il dispositivo con cui vengono svolte le misure di consumo carburante nella cella. Il funzionamento è basato sul **principio gravimetrico** che consente una misura diretta della massa consumata. Durante la calibrazione non è dunque necessaria l'acquisizione di temperatura e densità del carburante come per il principio volumetrico, sulla quale si produrrebbe un inevitabile errore di tolleranza.

L'utilità di questo strumento deriva dalla necessità di ottimizzazione di un motore soprattutto in fase di calibrazione nella quale, al variare dei parametri di input al gruppo termico, vediamo le variazioni di consumo.



Figura 8 Bilancia AVL 733s

Le funzioni possibili sono:

- Misure con intervalli start/stop: è possibile iniziare un intervallo di misura in qualunque fase del funzionamento della cella con intervalli limitati manualmente dai comandi start e stop;
- Misure singole o multiple con intervalli prefissati: tramite le impostazioni si definiscono intervalli di tempo fissi entro i quali vengono misurati i consumi di carburante. La misura può essere singola o l'insieme di più intervalli;

Inoltre sono disponibili misure intermedie (es. consumi istantanei), il consumo medio, massimo e minimo e la deviazione standard di misure multiple.

Il peso massimo processabile è la capienza massima del piatto di misura: 1800g.

La bilancia viene quindi riempita prima di iniziare le prove, mentre il consumo viene calcolato in base alla diminuzione nel piatto interno con un massimo di 150Kg/h per la configurazione standard. L'accuratezza caratteristica della misura è dello 0.1%.

In configurazione standard possono essere processati tutti i carburanti con una percentuale di etanolo inferiore al 15%; per passare a concentrazioni maggiori (fino all'etanolo puro) occorre sostituire le quattro connessioni idrauliche con apposite connessioni in acciaio inox (kit flex-fuel).

3.1.1 - Principi Operativi

Il sistema è composto dai seguenti componenti:

- Un **piatto di misura** (*measuring vessel*) che viene riempito dal carburante e sul quale viene misurato il consumo;
- Quattro tubi flessibili (*flexible tubes*) collegati al circuito esterno con le apposite connessioni idrauliche;
- Un peso che funge da **tara** (*tare weight*) con la funzione di bilanciare il piatto vuoto;
- Una **molla a balestra** (*blade spring*) che costituisce il fulcro del sistema di misura;
- Un **sensore capacitivo**;
- Uno **smorzatore** (*hydraulic damping device*).

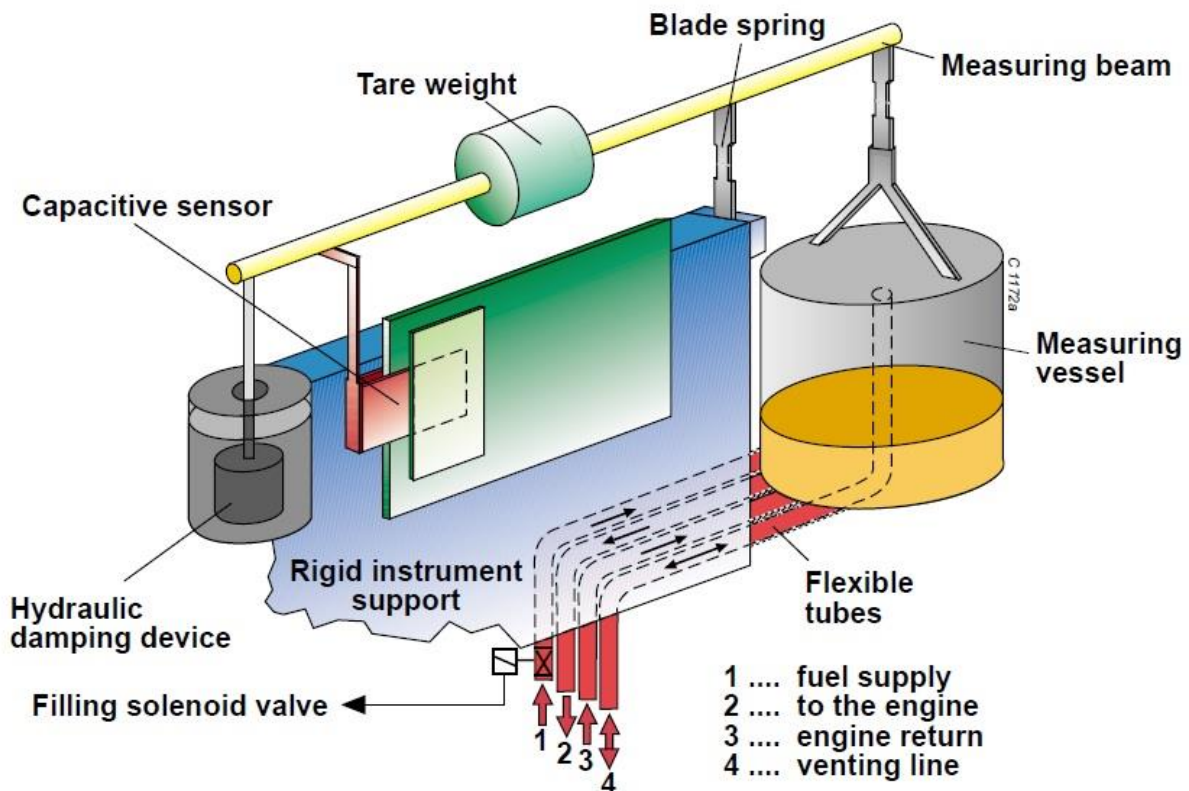


Figura 9 Componenti interni

Il carburante viene prelevato dal serbatoio esterno alla cella e arriva alla bilancia tramite la **linea di rifornimento** (*fuel supply*). Prima dell'ingresso nel dispositivo è presente un'**elettrovalvola a solenoide** (*filling solenoid valve*) con la funzione di regolare il riempimento del piatto. Tale valvola ha un doppio comando:

- Esterno fornito dall'utente tramite la funzione di *refill* che può essere attuato in ogni momento;

- Interno al sistema che regola sia la chiusura quando il piatto è pieno, sia l'apertura quando si oltrepassa la soglia minima di riempimento.

Il piatto è collegato al motore tramite un circuito chiuso che prevede due linee:

- Dal piatto al banco per fornire il carburante al rail e quindi agli iniettori;
- Dal motore al piatto per gestire il ricircolo del carburante fornito dalla pompa ma non utilizzato dal rail. Su questa linea è presente uno scambiatore controcorrente in modo che il fluido processato entri in bilancia nelle stesse condizioni di temperatura e densità con le quali è uscito, senza inficiare la misura.

Tale circuito viene denominato **circuito di misura** (*measuring circuit*).

Infine in uscita dal piatto abbiamo la **linea di sfiato** (*venting line*) necessaria per eliminare le bolle di vapore dal circuito.

La misura avviene tramite lo svuotamento del piatto: esso tramite la molla a balestra e l'asta di misura provoca un movimento del sensore capacitivo che tramuta il percorso svolto in voltaggio che sarà poi inviato al PC HOST come segnale digitale.

Il sensore non è sottoposto ad alcuno strisciamento od isteresi.

Per un utilizzo accurato occorre adottare alcune misure preventive sul circuito di misura:

- La temperatura del fluido interno deve essere costante ed in particolare uguale a quella fornita dalla linea di rifornimento per evitare cambi di peso specifico del fluido da un ciclo di riempimento all'altro;
- Il circuito va completamente privato di aria o bolle di vapore;
- Il volume geometrico del circuito di misura deve rimanere costante.

In particolare l'errore causato dalla perdita di peso dovuta alla temperatura può essere molto alto anche per piccole variazioni; la relazione della perdita di peso con la variazione di temperatura è data dalla seguente formula:

$$\Delta G = V \cdot \rho \cdot \alpha \cdot \Delta t$$

Dove V: volume geometrico del circuito di misura;

ρ : densità del fluido in esame;

α : coefficiente di dilatazione termica;

Δt : incremento medio della temperatura.

Il valore di ΔG viene tolto dal consumo effettivo del motore, generando così un errore di sottostima dei consumi netto su ogni misura.

3.1.2 – Circuito di Misura

L'importanza dell'isolamento del circuito di misura rispetto a temperatura e pressione è già stata spiegata; vediamo ora come procedere affinché le misure siano il più accurate possibili tramite alcune linee guida sull'installazione del sistema di misurazione.

Innanzitutto è buona norma tenere il circuito di misura il più breve possibile in modo da avere il minimo volume geometrico soggetto ad errore.

I seguenti accorgimenti vanno a correggere errori dovuti al **gradiente** di pressione e di temperatura.

Iniziamo con la **variazione di pressione**: vogliamo evitare perdite, problemi di ventilazione e depressioni locali (pericolose per la formazione di bolle di vapore):

- La resistenza idraulica nei tubi deve essere bassa;
- Le curve dei tubi devono essere morbide ed avere grandi raggi di curvatura;



Figura 10 Raggi di raccordo ottimali

- Mantenere le linee del circuito monotone discendenti o ascendenti; in particolare i vari componenti devono essere attraversati dal fluido dal basso verso l'alto per favorire una ventilazione ottimale;

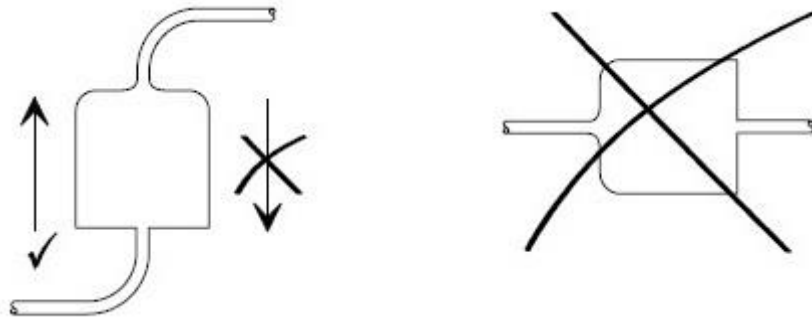


Figura 11 Schema dell'attraversamento dei componenti ottimale

- Occorre mantenere un flusso costante compreso fra 0,5 e 1 m/s per favorire il trasporto delle bolle ed evitare la loro generazione.
- Evitare bruschi cambi di sezioni che introdurrebbero vortici e distacchi della vena fluida generando bolle di vapore.



Figura 12 Esempio di formazione di vortici nei bruschi cambi di sezione

- Infine prima dell'installazione i tubi e i componenti andrebbero sciacquati con carburante pulito e soffiati con aria compressa.

Una possibile configurazione base ottimale è riportata in Figura 13.

Per evitare le **variazioni di temperatura media** invece occorre:

- Non disporre le linee parallelamente a fonti di calore (es. scarichi del motore);
- Isolare termicamente le linee;
- Installare un sistema di scambio termico per uniformare la temperatura del circuito di misura con quello di rifornimento e per correggere i gradienti dati da pompe e valvole elettroattuate;
- Una volta scelte le pressioni e temperature di utilizzo assicurarsi che in nessun punto venga raggiunta la pressione di vaporizzazione;

- Infine disporre il minor numero di valvole e rubinetti di drenaggio possibile ed in maniera saggia in modo da drenare solo il fluido necessario in caso, ad esempio, di cambio del filtro.

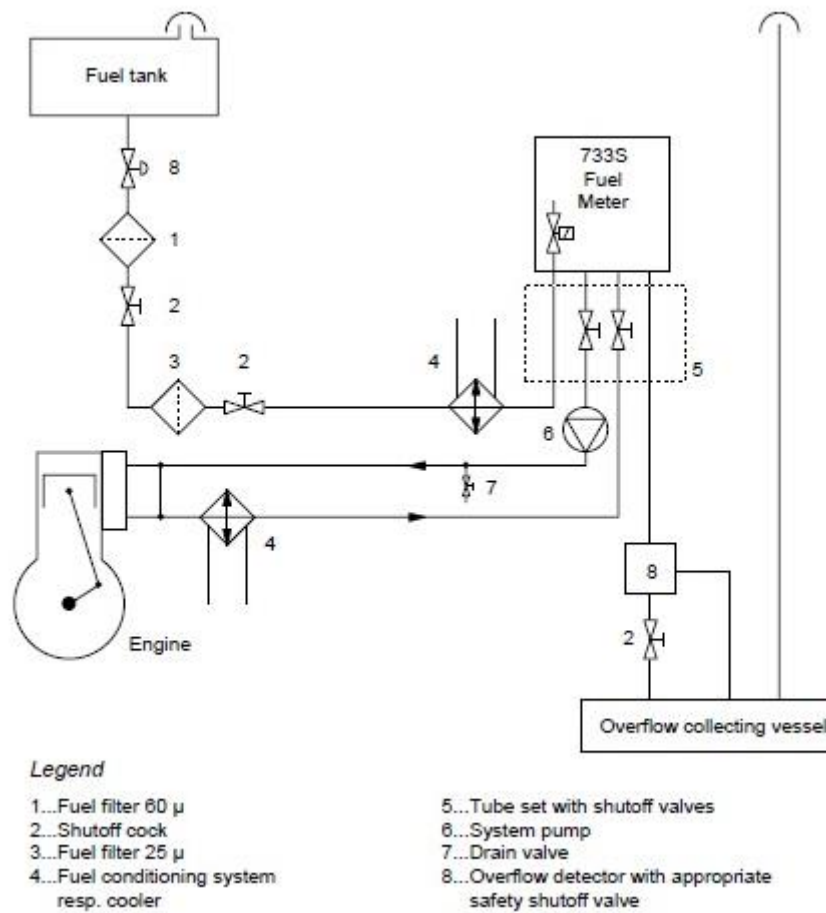


Figura 13 Schema di impianto base ottimale

3.2 – VI AVL_733

La bilancia è collegata al PC HOST con un cavo BV1714 tramite un'interfaccia seriale RS232. Il sistema di comunicazione può avvenire tramite due tipi di linguaggio: un **protocollo AK** e uno AVL 733. La nostra scelta ricade sul primo poiché solo con esso sono disponibili tutte le funzioni della bilancia ed inoltre è disponibile anche nella versione **Modalità di Servizio** (*service mode*) grazie alla quale è possibile comandarlo mediante terminale fornito da AVL.

Oltre alle funzioni già definite della bilancia sono disponibili anche:

- Calibrazione automatica;
- Controllo dell'accuratezza della misura automatico;
- Controllo delle valvole di riempimento e di bypass.

All'interno del PC HOST deve quindi essere presente un'interfaccia che dialoghi in modo seriale con la bilancia.

3.2.1 – Segnali di Input e Output

Prima della creazione dell'interfaccia occorre capire quali segnali vogliamo inviare e ricevere dal dispositivo.

Dividiamo quindi le funzioni utili in **Input** e **Output**.

Gli **Output** sono i segnali che provengono dalla bilancia e quindi le informazioni utili relative sia allo stato dell'apparecchio (connessione, stato, riempimento, ...) sia alla misurazione (consumi, livello di carburante, modalità di utilizzo, ...).

I parametri di **stato** sono:

- *Device Info*: indica il nome dell'apparecchio a cui si è connessi. Se la connessione tra Host e Bilancia avviene correttamente questo output fornisce il nome del device al quale si è connessi.
- Stato del device: la bilancia può essere connessa in *Manual*, stato nel quale si opera direttamente sulla macchina (ad esempio in fase di calibrazione) oppure in *Remote* dando la possibilità di agire da Host. Il segnale indica quindi quando è possibile fornire comandi al device.

- *Filling*: indica che la bilancia è in fase di riempimento. Durante questa operazione non è possibile eseguire misurazioni; tutti gli input vengono quindi interrotti o annullati.
- *Action*: descrive l'azione in atto.
- *Error*: tramite un codice ci indica i possibili errori di connessione, azione o controllo.

I dati di **misurazione** invece sono:

- *Filling level*: indica il livello di carburante presente nel piatto della bilancia.
- *Interval(s)*: fornisce il numero di prove nel caso di misurazioni multiple.
- *Consumption*: specifica il consumo istantaneo in kg/h e quello assoluto in g.
- *Measuring time*: indica la durata di una prova.
- *Max, min e standard deviation*: come ulteriori risultati abbiamo il consumo massimo, minimo e la deviazione standard riferita a prove multiple.

Gli **Input** sono tutti i comandi che vengono forniti al device per eseguire operazioni (*controls*), per richiedere dati (*interrogation*) o per impostare i parametri di controllo (*parameters*).

I **controlli** tipici sono:

- *Reset*: riporta il device allo stato subito successivo all'accensione.
- *Standby*: interrompe tutte le procedure in atto. In modalità standby il livello di carburante viene tenuto entro due livelli soglia ravvicinati (*Overflow e Threshold*) grazie a continui riempimenti di breve durata. In questo caso è possibile fare prove di lunga durata.
- *Refill*: aziona l'elettrovalvola per il riempimento del piatto di misura. Quest'operazione può avvenire automaticamente (quando si oltrepassa il valore minimo di carburante) oppure manualmente tramite l'azione di refill.
- *Bypass*: attiva il circuito di bypass che esclude la bilancia.

Le **interrogazioni** sono:

- *Start/stop*: avvia o termina l'inizio di una prova fornendoci i dati relativi al consumo.
- *Preselected Weight or Time*: avvia una misurazione con un tempo od un peso di carburante consumato preimpostato.

Infine i **parametri** impostabili sono relativi:

- Alla misura: numero di prove da fare (1-99), il tempo prescelto di misura in secondi (0.6-90000s) ed il peso prescelto di misura in grammi (1-1800 g); ad ogni reset i parametri di misura vengono riportati ai valori standard 1, 5, 50.

- Ai parametri di funzionamento: amplificazione del segnale al sensore capacitivo (1, 2, 4 o 8), il tempo medio del segnale del sensore in secondi (0.1-3s con massimo un decimale), l'intervallo di derivazione per valori istantanei in secondi (0.1-5s con massimo un decimale). I valori standard sono 1, 10, 0.5, 1.
- Ai parametri di sistema: *overflow/threshold* ad un'amplificazione di 1 in grammi (500-5000g); la percentuale di *overflow* per bloccare il riempimento (10-99%); ad ogni modifica di questi due parametri i cambiamenti genereranno un errore di non calibrazione del dispositivo; la percentuale di soglia minima (1-50%); la percentuale di soglia di riempimento in modalità Standby (0-100%); il coefficiente di temperatura in parti per milione su gradi centigradi (interi fra ± 5000); se il coefficiente cambia viene variata la relazione lineare fra il voltaggio e misura di carburante generando un errore di calibrazione; i valori standard sono 2000, 90, 5, 90, 1500.

Ora che sono stati definiti i segnali di ingresso ed uscita si può iniziare ad elaborare l'interfaccia.

3.2.2 – Interfaccia grafica

Tramite LabView è stata creata un'applicazione che rende possibile il dialogo con il device da Host: essa sarà divisa in *Control*, *Interrogation* e *Setting* in modo da essere il più simile possibile al modo di operare della macchina.

Per renderla intuitiva tale suddivisione avviene dividendo il programma in tre schede (TAB) distinte, una per ogni categoria.

Nella scheda Control si trovano sia gli Input che gli Output relativi alle azioni di controllo. In aggiunta l'input **EXIT** rende possibile lo scollegamento dal device e quindi l'interruzione del programma.

Gli input sono rappresentati dai pulsanti tramite i quali vengono inviati i comandi alla bilancia. Per quanto riguarda i 'controlli', i comandi *Reset*, *Standby*, *Refill* e *Bypass* avviano le funzioni già spiegate. Gli output possono essere digitali a led (accesi o spenti) oppure analogici nei quali è possibile leggere il dato numerico o il messaggio inviato dalla bilancia. In particolare la casella relativa agli errori legge il messaggio inviato dalla bilancia e lo decodifica in modo da fornire all'utente l'errore relativo.

I riquadri *Action* e *Device Info* forniscono rispettivamente il codice macchina dell'azione che stiamo svolgendo e la sigla del dispositivo a cui siamo collegati, che in questo caso sarà **733S Vx.xx** (al posto delle x ci saranno i numeri relativi alla versione del software AVL).

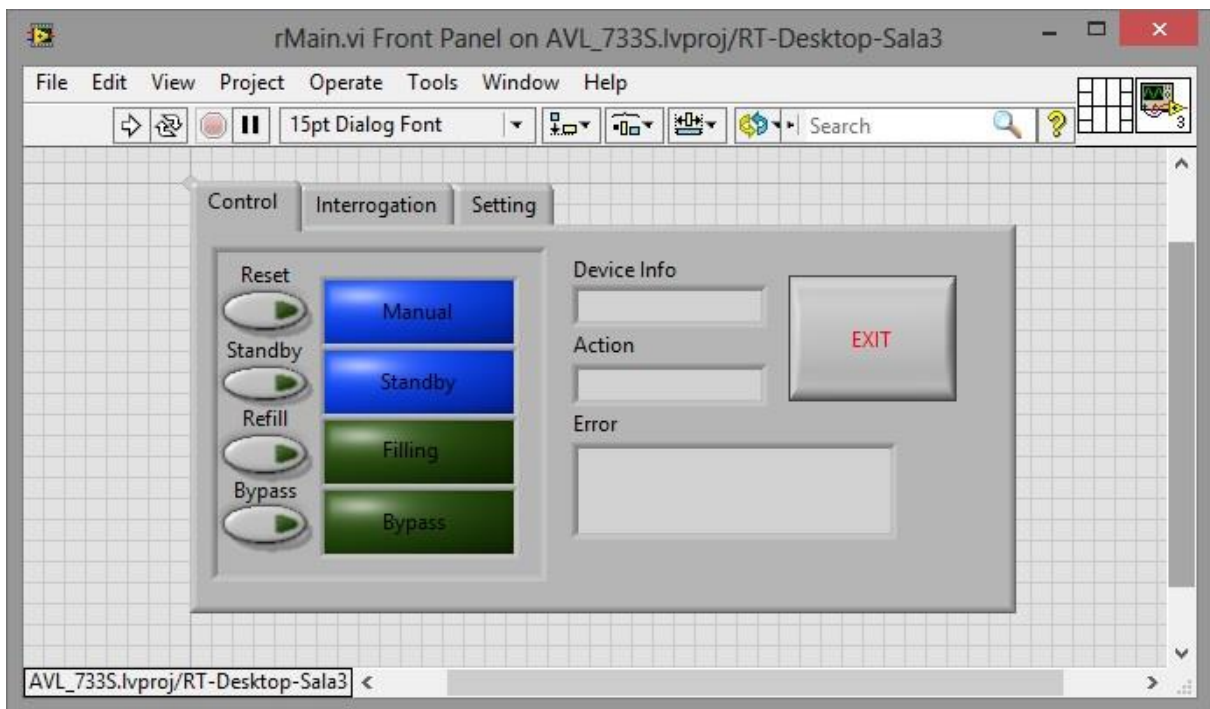


Figura 14 Scheda Control del pannello frontale

La scheda relativa alle interrogazioni presenta tutti i dati utili in fase di misurazione.

Nella parte sinistra sono raccolti tutti i dati relativi alle misure acquisite in modalità continua: tramite un ciclo continuo di interrogazioni la bilancia fornisce istante per istante i dati sul livello di carburante nel piatto e il consumo.

In particolare, la quantità disponibile viene indicata sia con una barra grafica numerata da 0 a 1800 in cui è possibile vedere qualitativamente il livello (*Filling level*), sia con un dato numerico per una misura precisa in g.

Essendo dati in continuo tali valori cambiano istantaneamente durante il funzionamento.

Nel riquadro di destra vengono forniti gli output relativi alle misure. I dati sono diversi a seconda della scelta della modalità di analisi fra *Preselected & Start/ Stop* oppure *Preselected Weight & Time*.

Nella prima vengono forniti i valori misurati durante una prova nella modalità Start/Stop in cui l'intervallo viene indicato manualmente con l'apposito comando. Gli output numerici sono:

- *# of evaluation*: ossia il numero di prove effettuate, utile nel caso di test multipli.
- *Measuring Value*: il consumo orario in kg/h.
- *Measuring Time*: il tempo di durata della prova.
- *Consumption*: consumo assoluto in g.

Nella seconda modalità i risultati sono relativi a prove con un tempo o un peso di carburante consumato preimpostati. I dati di uscita sono:

- *# of evaluation*: come nella precedente modalità.
- *Mean Value*: ossia il consumo medio durante la prova, in kg/h.
- *Max, min e Standard deviation*: relativi alla prova.

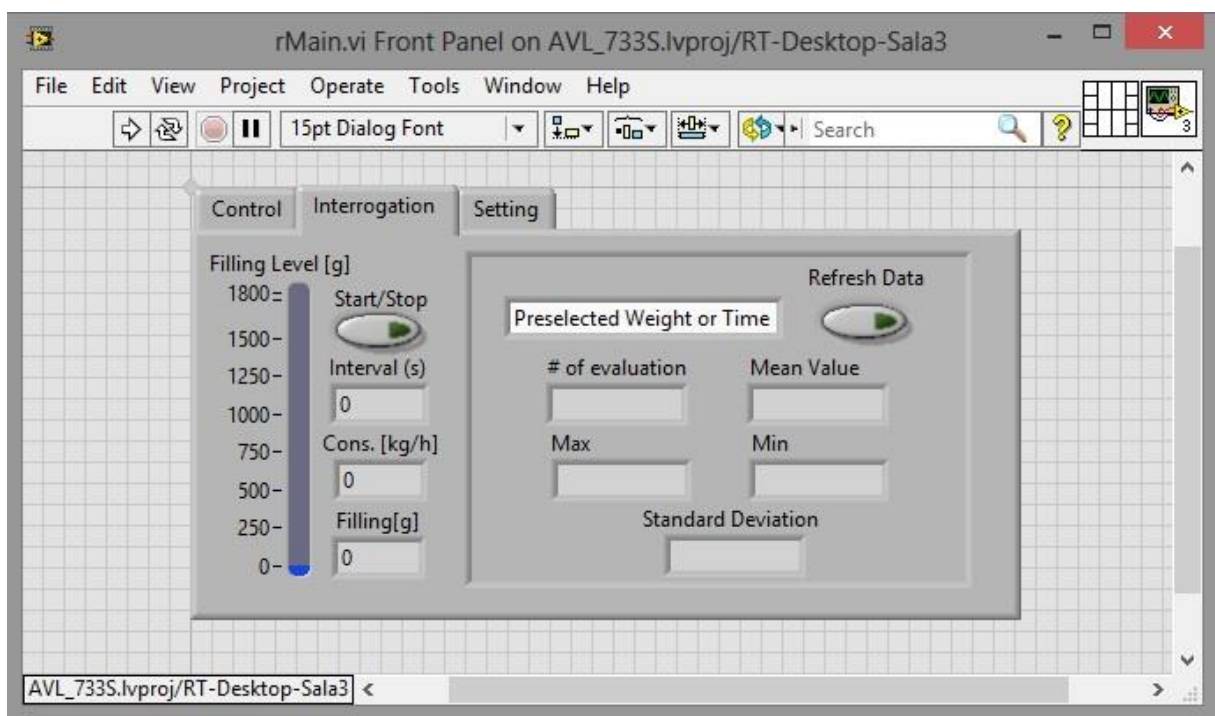


Figura 15 Scheda Interrogation del pannello frontale

L'input **Refresh Data** consente di interrogare nuovamente la bilancia con l'ultima modalità scelta.

Nell'ultima scheda (*Setting*), relativa ai parametri di impostazione, si ha la possibilità di agire sulle impostazioni della bilancia suddivise in:

- *Measurment*
- *Main*
- *System*.

Nella parte alta sono riportati tre input che agiscono sulla modalità selezionata:

- *Standard Values*: invia alla bilancia i valori standard presenti sul manuale.
- *Enable Chngement*: una volta modificati i valori in remoto con questo input li si invia alla bilancia che provvede a cambiarli internamente. Fino a che non vengono inviati i dati rimangono quelli correnti in uso.
- *Current Value*: mostra i valori attuali di ciascuna categoria presenti nella bilancia.

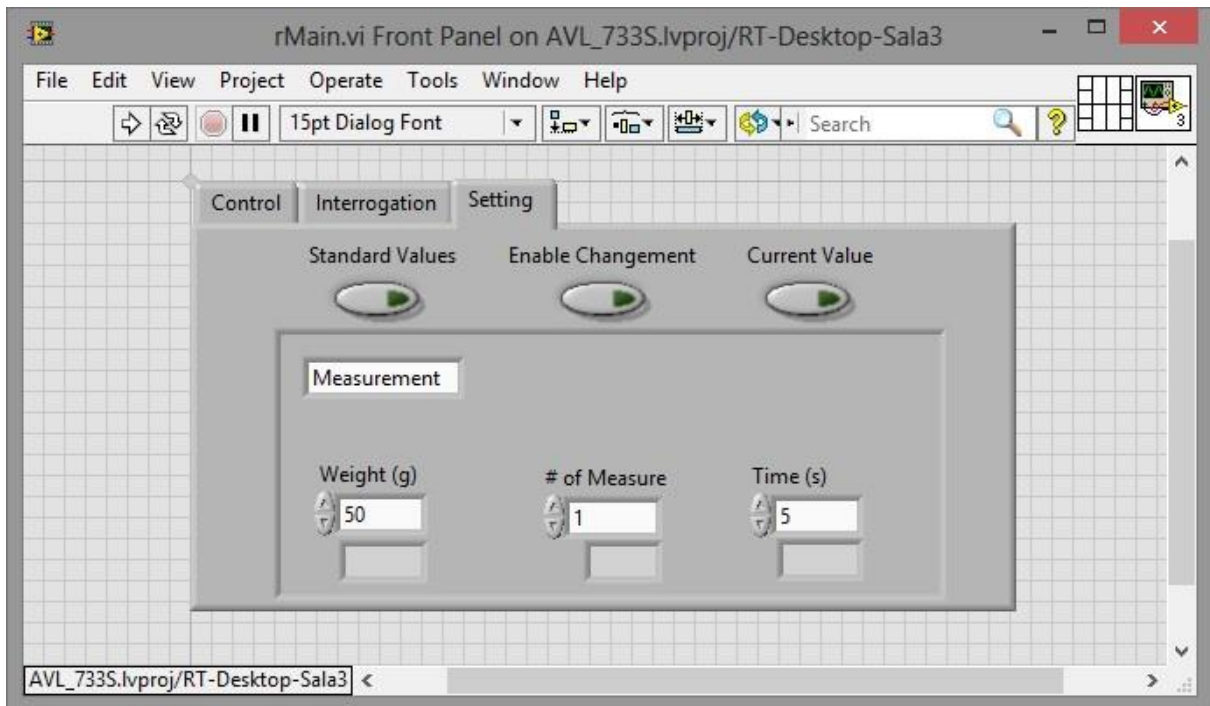


Figura 16 Scheda Setting del pannello frontale

Tramite questa semplice ed intuitiva interfaccia è dunque possibile eseguire tutte le funzioni input ed output descritte in precedenza.

3.2.3 - AK-Protocol

Il protocollo utilizzato per dialogare con il device, come già descritto, è quello standard AK fornito dal manuale AVL. La comunicazione avviene con una modalità **master-slave**, in cui l'utente ordina un'azione o richiede un dato e il dispositivo obbedisce o informa del possibile errore.

La sequenza delle operazioni è sempre la stessa:

- Invio del codice di comando;
- Lettura del codice da parte del dispositivo;
- Azione;
- Risposta del dispositivo;
- Ricezione e codifica del dato da parte dell'interfaccia.

Tutti i messaggi scambiati fra dispositivo e Host vengono trasformati in formato **ASCII**. Le stringhe sono composte da *n byte* a seconda della loro funzione: in generale sono suddivise in **comandi e risposte**.

I comandi (*command message*) sono i messaggi inviati alla bilancia dal terminale.

La composizione totale del messaggio è la seguente:

Byte	Function	
1	<STX>	
2	Don't care byte	
3...6	Function code SXXX ...Control commands AXXX ...Interrogation commands EXXX ...Setting commands	
7	Blank	Channel information may be omitted
8	K ...Identification of channel number	
9	0 ...Channel number	
10	Blank (only if followed by a variable data block)	
11...n-1	Variable data block (only for setting commands and interrogation commands AERG and ABEW)	
n	<ETX>...End of command	

Figura 17 Composizione dei Command Message

Il primo e l'ultimo byte (<**STX**>, <**EXT**>) indicano l'inizio e la fine del messaggio nel linguaggio ASCII.

Il secondo è il *don't care byte*.

Dal terzo al sesto abbiamo il vero e proprio **Codice Funzione** (*function code*) formato da 4 byte; i comandi possono essere divisi in 3 categorie a seconda della lettera iniziale del loro Codice Funzione:

- Controlli: **Sxxx**;
- Interrogazioni: **Axxx**;
- Impostazioni: **Exxx**.

I byte dal settimo al decimo sono riservati alle informazioni di canale che possono essere omesse. Questi byte non rientrano nel nostro interesse e pertanto saranno saltati.

Infine dall'undicesimo fino al byte finale abbiamo il **blocco delle variabili** (*variable data*) disponibile sono per alcune interrogazioni e per le impostazioni.

I formati di tali dati sono:

- Identificazione di un dispositivo definibile con un massimo di 40 caratteri ASCII;
- Valori numerici del tipo floating point di lunghezza variabile e numeri interi (in questo caso la parte decimale può essere omessa); il segno può essere messo solo davanti ai numeri negativi.

Tra un dato e l'altro occorre mettere uno spazio di separazione.

I valori senza alcun significato fisico non saranno valutati.

Le risposte (*acknowledge message*) sono invece i messaggi che la bilancia invia al terminale; al loro interno sono contenuti i dati richiesti dal comando. Come per i comandi il primo e l'ultimo byte sono riservati all'inizio e fine messaggio, mentre il secondo è occupato dal *don't care byte* copiato dal comando.

Anche il codice funzione è il medesimo per indicare il comando da cui deriva la risposta.

L'ottavo byte è occupato dallo **stato dell'errore** (*error status*): se l'operazione non crea problemi il byte è 0 altrimenti esso verrà incrementato da 1 a 9 per ogni cambiamento fino a che tutti gli errori non siano risolti.

Byte	Function
1	<STX>
2	Don't care byte
3..6	Funktion code (identical with the command)
7	Blank
8	Error status (0... no error)
9	Blank (only if followed by a variable data block)
10...n-1	Variable data block (depending on the function code)
n	<ETX>...End of acknowledgement

Figura 18 Composizione degli Acknowledge Message

Se il comando non inizia con <STX> o il codice funzione è sconosciuto la risposta avrà le seguenti caratteristiche:

- Il *don't care byte* sarà sostituito con uno spazio;
- Il codice funzione sarà sostituito da **????**;
- Il blocco delle variabili sarà omissso.

Il blocco delle variabili sarà diverso a seconda del comando.

Per un controllo (**Sxxx**) se il comando è valido il blocco verrà omissso; in caso contrario il comando non verrà eseguito e la ragione del rifiuto sarà indicata nel blocco con:

- **K0 OF** la bilancia non è in Modalità Remota;
- **K0 BS** il comando non è permesso nello stato attuale della macchina;
- **K0 NA** la funzione non può essere momentaneamente eseguita.

Per un'interrogazione (**Axxx**) se il comando è valido i dati richiesti saranno scritti nel blocco (eventualmente separati da uno spazio se multipli); i dati possono essere floating, integer o status (una stringa di 4 byte ASCII); in caso contrario i dati non validi saranno rimpiazzati con un # (o anticipati da # se validi con riserva).

Per un'impostazione (**Exxx**) se il comando è valido il blocco verrà omissso; in caso contrario il comando non verrà eseguito e la ragione del rifiuto sarà indicata nel blocco con:

- **K0 OF** la bilancia non è in Modalità Remota;
- **K0 BS** il comando non è permesso nello stato attuale della macchina;
- **K0 DF** i dati inviati al dispositivo non appartengono al range permesso.

Tutti questi codici **K0 xx** sono decodificati nella casella Error dell'interfaccia in cui compare semplicemente il motivo dell'errore senza il codice.

Di seguito sono riportati i codici delle sole funzioni input ed output utilizzate all'interno del VI.

I comandi di controllo (**Sxxx**) permettono di svolgere le principali azioni:

- **SRES**: reset, reimposta il dispositivo nello stesso stato dell'accensione; dopo l'inizializzazione vengono ristabilite le funzioni **SMAN**, **SMES**, **SAVZ** e **SBVZ**; il comando è sempre disponibile;
- **SREM**: porta la bilancia in Modalità Remota; il comando è sempre disponibile;
- **STBY**: annulla e interrompe le procedure in corso ed attiva lo Standby Mode; il comando è disponibile solo il Remoto;
- **SMES**: annulla e interrompe le procedure in corso ed attiva la modalità di misura; il comando è disponibile solo il Remoto;
- **SINT**: inizializza un intervallo di misura (start/stop); il comando è disponibile solo in Remoto negli stati **SMES SAVZ** o **STBY SAVZ**; quando si raggiunge il livello minimo nel piatto l'intervallo viene annullato e vengono ristabiliti gli stati **SMES** o **STBY**;
- **SMIS**: stoppa l'intervallo di misura; il comando è permesso solo in Remoto nello stato **SINT**; vengono ristabiliti gli stati **SMES** o **STBY**;
- **SMZV**: inizializza la procedura di misura con intervallo preselezionato; il comando è disponibile solo in Remoto negli stati **STBY** o **SMES** che verranno ristabiliti dopo il completamento della procedura di misura;
- **SMGV**: inizializza la procedura di misura con peso preselezionato; il comando è disponibile solo in Remoto negli stati **STBY** o **SMES** che verranno ristabiliti dopo il completamento della procedura di misura;
- **SAVA**: apre la valvola di riempimento; il comando è disponibile solo in Remoto eccetto negli stati **SATK**, **SKCH** e **SRES**; se il livello nel piatto è maggiore del massimo la valvola rimane chiusa; le misure correnti vengono interrotte;
- **SAVZ**: chiude la valvola di riempimento; il comando è disponibile solo in Remoto eccetto negli stati **SATK** e **SKCH**; se il livello nel piatto è minore di 0 gr o della

soglia di riempimento la valvola rimane aperta; questo stato viene raggiunto solo quando la valvola è completamente chiusa e dopo l'azione di pompaggio;

- **SBVA**: apre la valvola di bypass escludendo la bilancia; il comando è disponibile solo in Remoto eccetto in **SATK** o **SKCH**;
- **SBVZ**: chiude la valvola di bypass includendo la bilancia nel circuito; il comando è disponibile solo in Remoto eccetto in **SATK** o **SKCH**;

I principali comandi d'interrogazione (**Axxx**) servono a richiedere i dati delle nostre misure:

- **AKEN**: identifica il dispositivo con 733S Vx.xx;
- **ASTZ**: indica lo stato del dispositivo;
- **AWRT**: indica i valori attuali nel blocco:
 1. livello carburante nel piatto (g);
 2. valore istantaneo del consumo in Kg/h con un massimo di 3 cifre decimali;
 3. numero della misura o # se non sono state effettuate misure;
 4. consumo (g);
 5. tempo medio (s);
- **AERG n**: risultati numerici con una misura start/stop:
 1. il numero di prove da fare;
 2. il valore del consumo in kg/h fino al terzo decimale;
 3. il tempo di misura in secondi fino al primo decimale;
 4. il consumo in g fino al secondo decimale.
- **ABEW n**: risultati numerici con una misura a tempo o peso prestabilito:
 1. il numero di prove da fare;
 2. il valore del consumo medio in kg/h fino al terzo decimale;
 3. il valore del consumo massimo in kg/h fino al terzo decimale;
 4. il valore del consumo minimo in kg/h fino al terzo decimale;
 5. il valore della deviazione standard in kg/h fino al quarto decimale.
- **AMPA**: parametri di sistema corretti definiti in EMPA:
 1. il numero di prove da fare;
 2. il tempo prescelto di misura in secondi (s);
 3. il peso prescelto di misura in grammi (g);
- **AHPA**: parametri di sistema corretti definiti in EHPA:

1. l'amplificazione;
 2. il tempo di pompaggio dopo il riempimento (s);
 3. il tempo medio del segnale del sensore (s);
 4. l'intervallo di derivazione per valori istantanei (s);
- **ASPA:** parametri di sistema corretti definiti in ESPA:
 1. sovrafflusso/soglia ad un'amplificazione di 1 (g);
 2. la percentuale di sovrafflusso per bloccare il riempimento;
 3. la percentuale di soglia minima;
 4. la percentuale di soglia di riempimento in modalità Standby;
 5. il coefficiente di temperatura in parti per milione su gradi centigradi;

I comandi d'impostazione (**Exxx**) servono a personalizzare i parametri di sistema e misura:

- **EMPA:** serve a definire nel blocco variabili i parametri di misura; il comando è disponibile solo in Remoto eccetto negli stati **SMZV**, **SMGV** e **SINT**; nel blocco vanno indicati 3 dati numerici: il numero di prove da fare (1-99), il tempo prescelto di misura in secondi (0.6-90000s) ed il peso prescelto di misura in grammi (1-1800g); ad ogni reset i parametri di misura vengono riportati ai valori standard 1 5 50;
- **EHPA:** serve a definire i parametri principali; il comando è disponibile solo in Remoto eccetto negli stati **STBY** o **SMES**; nel blocco variabili occorre indicare 4 dati numerici: l'amplificazione (interi 1, 2, 4, 8); ad ogni cambio verrà assunta un'amplificazione lineare generando un errore di non calibrazione del dispositivo; il tempo di pompaggio in secondi (1-300s con massimo un decimale); il tempo medio del segnale del sensore in secondi (0.1-3s con massimo un decimale); l'intervallo di derivazione per valori istantanei in secondi (0.1-5s con massimo un decimale); i valori standard sono 1 10 0.5 1.
- **ESPA:** serve a definire i parametri di sistema; il comando è disponibile solo in Remoto eccetto negli stati **STBY** o **SMES**; devono essere definiti i seguenti 5 dati numerici: sovrafflusso/soglia ad un'amplificazione di 1 in grammi (500-5000g); la percentuale di sovrafflusso per bloccare il riempimento (10-99%); ad ogni cambio dei primi due parametri verrà i cambiamenti genereranno un errore di non calibrazione del dispositivo; la percentuale di soglia minima (1-50%); la percentuale di soglia di riempimento in modalità Standby (0-100%); il coefficiente di temperatura in parti per milione su gradi centigradi (interi fra ± 5000); se il coefficiente cambia viene variata la

relazione lineare fra il voltaggio e misura di carburante generando un errore di calibrazione; i valori standard sono 2000 90 5 90 1500.

3.2.4 – *rMain*

L'interfaccia visibile dall'utente è programmata graficamente tramite uno schema a blocchi, ossia il linguaggio di **LabView**. In tale schema sono presenti tutte le operazioni logiche che il programma deve svolgere per funzionare correttamente.

La programmazione avviene in modalità **Real Time** (RT): in questo modo è possibile dialogare in modo continuo con il device tramite azioni ripetute nel tempo. Il loop temporale viene eseguito con una frequenza di 1 kHz.

All'avvio l'applicazione svolge una sequenza di azioni (VISA Serial, AKEN, SREM e SMES) tali da permettere il collegamento con il device e da fornirci le informazioni iniziali della scheda Control (Device Info, Error, Manual/Remote, Standby, ...) presente nell'interfaccia.

Lo schema è programmato tramite una struttura *case* nella quale a seconda dell'input inviato ci si sposta da un caso all'altro. Gli input non sono altro che i comandi che noi forniamo al sistema.

I casi della struttura sono:

- Default;
- Reset;
- Refill;
- Start/Stop;
- Parameters;
- Refresh;
- Interrogation.

Finita l'inizializzazione, il VI si porta al *case 0*, ossia il caso di **Default**: esso è il principale nel quale l'applicazione si ripete in loop continuo fino a che tramite l'input EXIT non ci si disconnette dal device. Al suo interno si richiedono continuamente alla bilancia le condizioni di stato (relative alla scheda Control) ma anche Filling Level, Consumption e Filling (relativi alla scheda Interrogation). In questo modo le informazioni principali sono sempre aggiornate in tempo reale.

Nella pagina successiva è riportato lo schema a blocchi del caso Default.

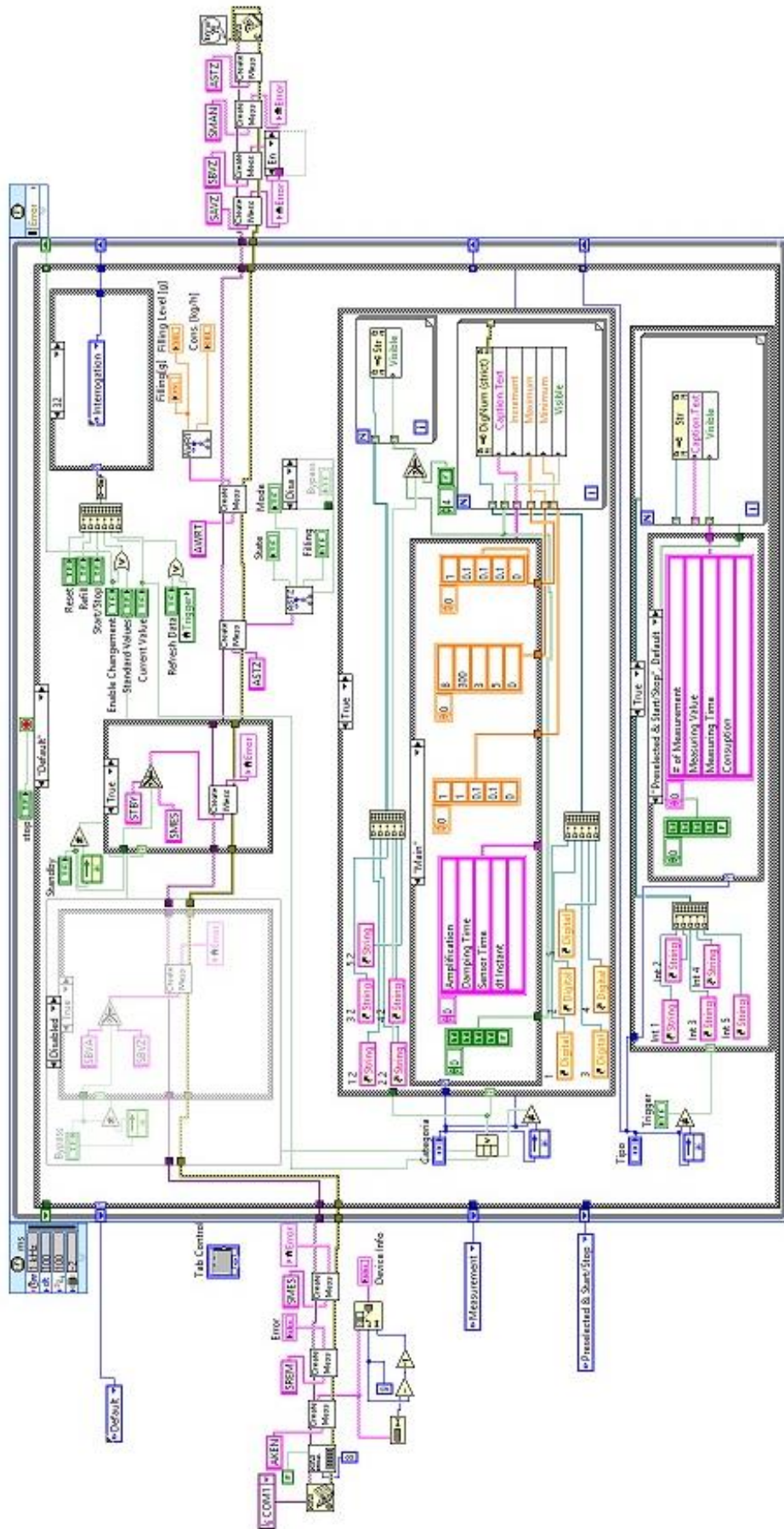


Figura 19 Schema a blocchi del caso Default

A questo punto a seconda dell'input che noi andiamo a selezionare nell'interfaccia grafica l'applicazione si sposta tramite un case selector costituito da un array di segnali logici ad ognuno dei casi elencati sopra.

All'interno dei casi viene svolta l'azione richiesta (es. Reset) con la modalità master-slave descritta in precedenza ed una volta terminato il task si ritorna a Default. La frequenza altissima delle operazioni consente di fatto una minima perdita delle informazioni richieste in continuo nel case 0, mentre si eseguono gli altri casi.

Il VI **rMain** è quindi l'essenza stessa del programma. Al suo interno tuttavia sono presenti dei **subVI**, ossia sottoprogrammi dipendenti dal principale che svolgono compiti differenti. Essi hanno una doppia valenza:

- Evitare parti di codice ridondanti: può capitare infatti che certe azioni vengano ripetute in più case, quindi creando un programma esterno che svolge tali funzioni è sufficiente richiamarlo ogni volta senza dover riscrivere il codice.
- Ottimizzare lo schema a blocchi: essendo una programmazione grafica ogni operazione occupa fisicamente spazio con linee, funzioni, indicatori, ecc. Risulta quindi comodo (ed esteticamente migliore) semplificare il codice con programmi esterni che occupano il minimo spazio.

3.2.5 – SubVI

I SubVi sono utili al funzionamento sono:

- Create message;
- Interrogation;
- ASTZ message;
- AWRT message;
- Create Main/System/Setting.

Create message è il principale subVI di tutta l'applicazione. Esso infatti consente sia di inviare messaggi alla bilancia che di decodificare gli errori contenuti nella risposta del device. Lo schema a blocchi è riportato in Figura 20.

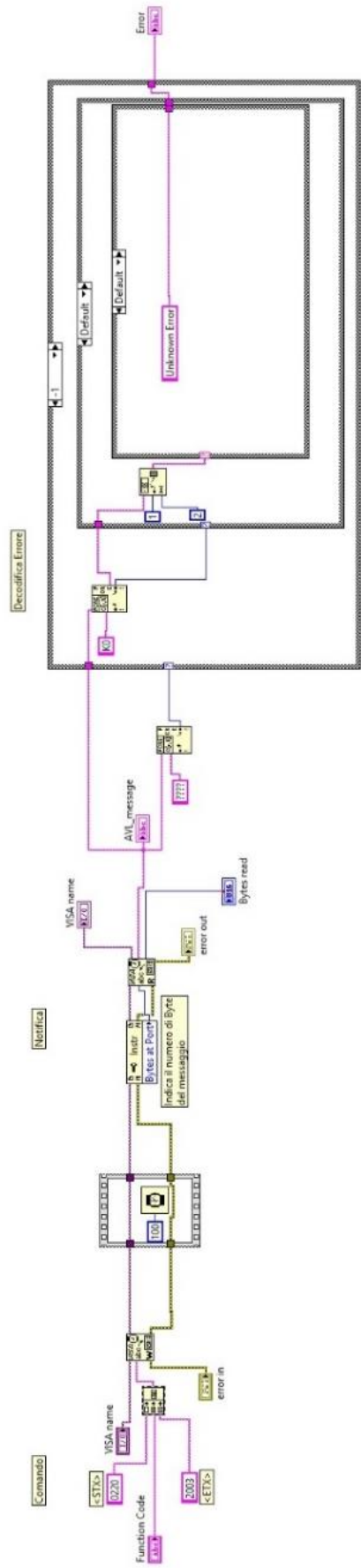


Figura 20 Schema a blocchi del subVI Create Message

In ingresso al programma abbiamo una stringa di 4 caratteri contenente il codice funzione che vogliamo inviare alla bilancia.

Nella prima parte, alla stringa fornita, vengono aggiunti tramite un concatenate strings i codici **STX** ed **ETX** in formato numerico ASCII: rispettivamente 0220 e 2003.

Ora che il messaggio è completo si inizia la comunicazione con la macchina che avviene tramite funzioni **VISA** integrate in LabView: si utilizza *Write* per compilare e *Read* per leggere le risposte. Tra comando e ricezione è necessario un intervallo di tempo in cui la macchina acquisisce la richiesta, elabora il dato e invia la risposta. L'attesa è ottenuta con un temporizzatore impostato a 100 ms. Tale valore è ottenuto sperimentalmente: ci si è resi conto che se il delta è minore non sempre si riesce a ricevere la risposta in tempo, mentre se superiore provoca problemi di sincronia con il loop del rMain.

Nell'ultima parte la decodifica dell'errore è realizzata nel modo più semplice possibile: si ricerca nella stringa la sigla **K0** (identificativa del codice di errore) e si leggono i due caratteri successivi (distaccati da uno spazio) che indicano il tipo di errore.

Questi due caratteri sono l'ingresso di un *case selector* che in output ha una stringa con la codifica a seconda del caso. In aggiunta è presente la decodifica della risposta *????* in caso di funzione sconosciuta, anche se, fornendo il codice automaticamente con una stringa e non da utente, non dovrebbe presentarsi questo caso.

Il secondo SubVI fondamentale è **interrogation**: esso si occupa di leggere i dati dopo una misurazioni e li trasforma in un array stringa che viene poi spezzato all'interno di rMain dove i vari dati sono inviati al proprio indicatore. Lo schema a blocchi è riportato in Figura 21.

In ingresso si hanno due stringhe: la prima è quella contenente il messaggio dalla bilancia con tutti i dati, la seconda è formata dai 4 caratteri del codice di interrogazione (**ASPA, AHPA, AMPA, ...**) così da non renderla specifica ad un solo tipo di richiesta. Proprio perché vogliamo rendere generale questo SubVI, rispetto a Create Message, la decodifica avviene in maniera differente.

Innanzitutto, tramite un *Search or Split String*, si cerca all'interno del messaggio la stringa fornita in input ed si inviano al resto del programma solo i caratteri successivi; tramite questo accorgimento vengono svolte due importanti funzioni: si verifica, tramite il codice funzione, che il messaggio di risposta sia quello desiderato (in caso contrario si invia un errore) e si semplifica la decodifica, agendo solo con la parte seguente del messaggio.

Una volta isolata la stringa contenete i soli dati, tramite un ciclo for, si legge carattere per carattere il messaggio. Ogni dato è diviso da uno spazio quindi, per generare il nostro array di output in cui ogni elemento è un dato, basta tramite una case structure, aumentare l'indice del vettore ogni volta che legge uno spazio.

Come ultimo accorgimento si correggerà nel messaggio, prima della decodifica, l'eventuale presenza della stringa # # # (indicazione della scorrettezza di una misura) con ###, così da evitare errori di indice nell'array di output.

Questo SubVI è utilizzato per mostrare i dati nella scheda Interrogation dell'interfaccia.

I SubVI **ASTZ** e **AWRT** sono invece i programmi tramite cui si fa l'analisi in continuo nella scheda Default. Funzionano in maniera analoga a Interrogation ma in output non hanno un array stringa contenente i dati, bensì i dati numerici (AWRT) sulla quantità di carburante o i segnali logici (ASTZ) sullo stato della bilancia.

In questo caso le stringhe di risposta sono predefinite (ad esempio SMES per indicare che la bilancia non è in standby) e quindi la decodifica avviene semplicemente andando a cercare ciò che ci interessa all'interno del messaggio. Nelle pagine seguenti sono riportati i due schemi a blocchi rispettivamente di ASTZ e AWRT nelle Figure 22-23.

Infine i vari **Create Main/System/Setting** sono simili a Create Message, con la differenza che nella stringa inviata, oltre al codice funzione (rispettivamente **EHPA/ESPA/EMPA**), si concatena alla stringa anche il valore numerico convertito dell'impostazione da modificare. Questi tre SubVI sono utilizzati per modificare le impostazioni della bilancia, sulle quali si agisce nella scheda di Setting.

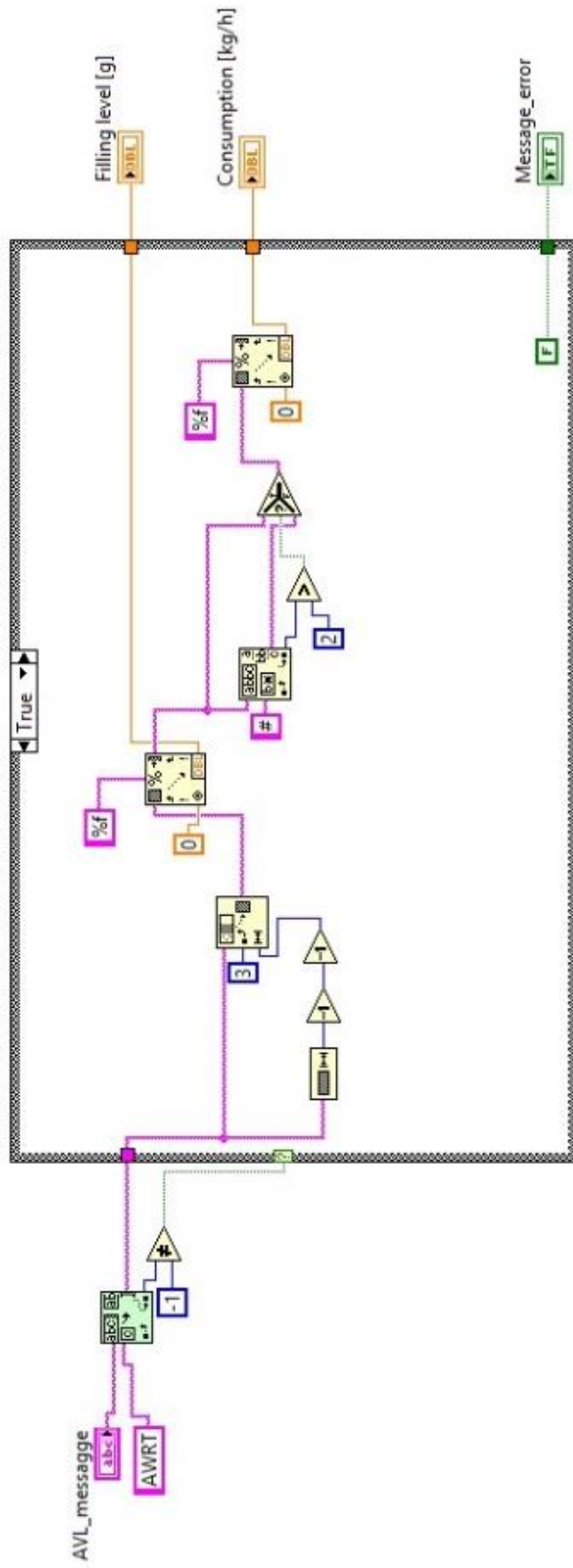


Figura 23 Schema a blocchi del SubVI `avl_awrt_mess_default`

3.3 - Altre Applicazioni

Il VI della bilancia è stato programmato anche per un'applicazione specifica per il banco prova n.2 dell'hangar di Forlì sul quale è installata una turbina. Di seguito viene riportata l'interfaccia utente:

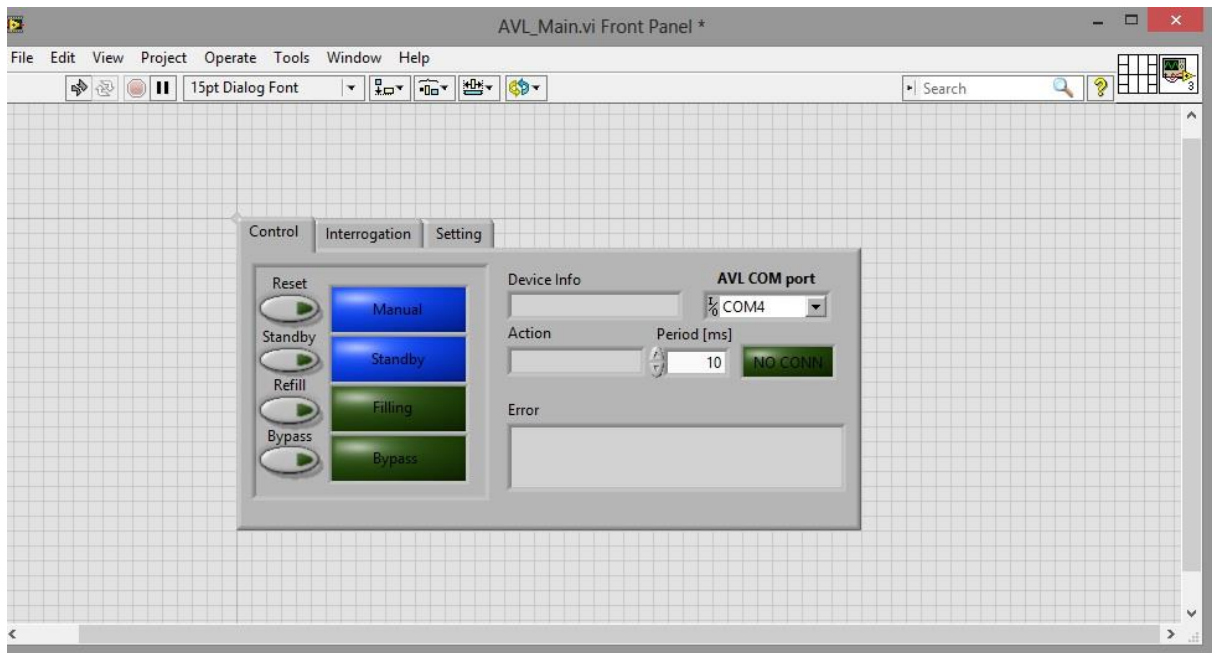


Figura 24 Pannello frontale dell'interfaccia del programma per il controllo Turbina

Nel pannello le differenze principali sono:

- Poter scegliere la porta **COM** tramite la quale si fa il collegamento al dispositivo: nella versione della tesi è invece impostata la COM 2 di default per ragioni di standardizzazione del modulo rack.
- La possibilità di scegliere il **periodo** di tempo del *ciclo loop* con il quale rMain si aggiorna. Per lo stesso motivo del punto precedente nel programma della tesi questo dato è fisso e non si può cambiare.

Nello schema a blocchi invece la differenza è che certi dati output sono inviati all'esterno del programma tramite variabili globali così di poter essere integrato in un sistema più complesso.

4 – Smoke Meter AVL 415S

4.1 - Dispositivo

Lo **Smoke Meter AVL 415s** è il dispositivo con cui vengono svolte le misure di inquinamento dei gas di scarico del motore montato al banco.

La misura avviene prelevando un certo volume di gas dallo scarico e forzando il suo passaggio attraverso un filtro di carta. L'annerimento del filtro causato dai gas di scarico viene quindi misurato tramite un rifrattometro da cui deduciamo la quantità di fuliggine, data dal carbonio, presente.



Figura 25 AVL 415s

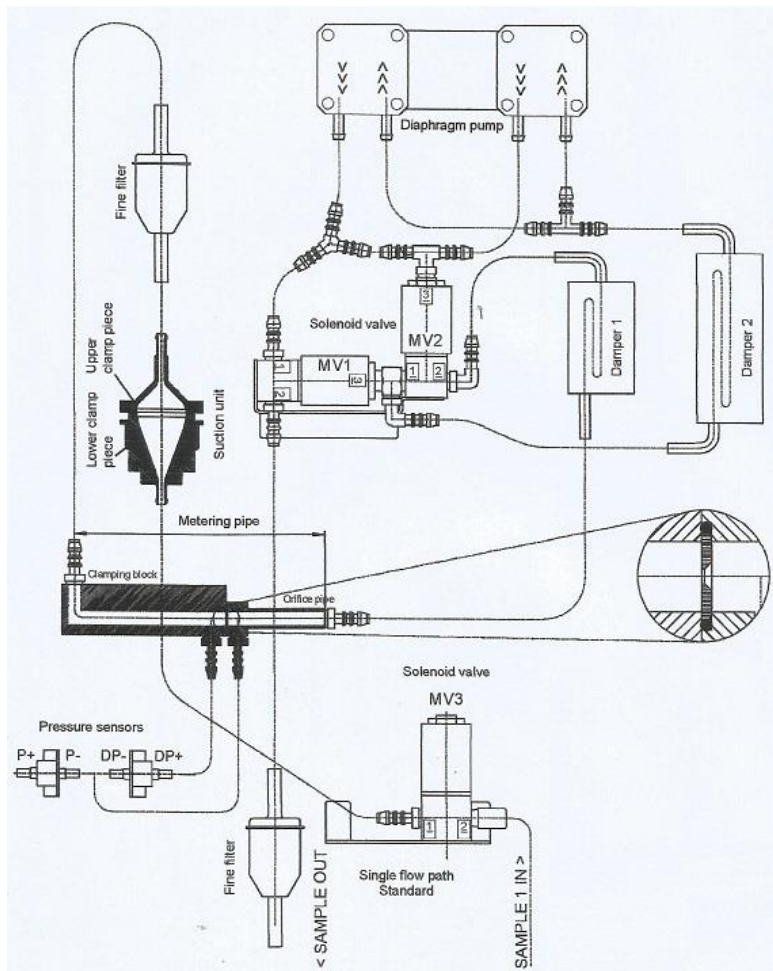
Nell'immagine è possibile vedere l'interno del dispositivo di misura. Il filtro di carta è arrotolato all'interno sia per garantire un certo numero di misure sia per una certa facilità di sostituzione.

Le misure possibili sono:

- Singole o multiple: partendo dai parametri standard si procede al ciclo di misura che dura qualche secondo; si può eseguire una misura singola o cicli multipli uno successivo all'altro.
- Con tempo o volume prescelti: in questo caso si fissa una delle due variabili fra durata della misura o volume predefinito. In entrambi i casi si misura l'annerimento che, come vedremo nei principi operativi, può essere differente a seconda del tempo di esposizione o del volume di gas aspirato.

Infine è possibile determinare la presenza media di fuliggine nei fumi.

Rispetto al modello AVL 415, la versione "S" prevede l'integrazione di un Heating Option che evita la formazione di condensa sul cilindro contenente il volume di prova e un sistema di prova più semplice con conseguente incremento della durata del filtro di carta.



Nell'immagine a fianco è riportato lo schema dell'impianto di circolazione dei gas di scarico all'interno dello smoke meter.

All'interno dell'area ingrandita viene fatto scorrere il filtro che entra così in contatto con la fuliggine che andiamo a misurare.

Figura 26 Schema del circuito dei gas aspirati

4.1.1 - Principi Operativi

Come introdotto nel capitolo precedente la misura degli inquinanti avviene forzando il passaggio di un certo volume di gas di scarico attraverso un filtro.

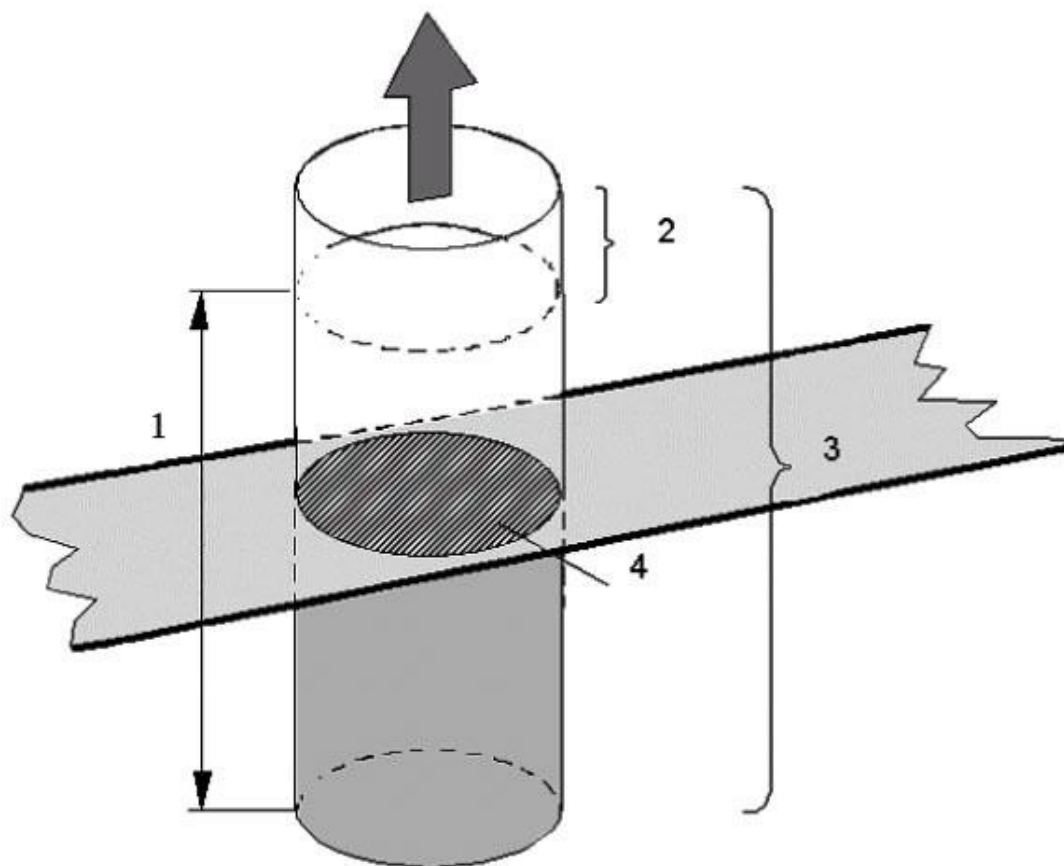


Figura 27 Schema dei volumi aspirati

Il volume di misura è quindi il principale parametro di input del sistema; per semplificare il procedimento viene introdotta la lunghezza effettiva. Tale parametro è legato alla normativa ISO DP 10054 che la definisce come:

$$Lunghezza\ Effettiva = \frac{\text{Volume di Misura} - \text{Volume Nocivo} - \text{Perdite}}{\text{Area del Filtro}}$$

I volumi presenti nella formula sono visibili nella figura:

1. Volume relativo alla lunghezza effettiva;
2. Volume nocivo;
3. Volume totale di gas aspirato;
4. Area del filtro.

Il volume nocivo è quella porzione di gas che va dalla sonda fino al filtro riempita di aria pulita e che non partecipa all'annerimento del filtro. Tramite una depurazione, preventiva alla misura, si può avere la certezza che non sia presente fuliggine all'interno tale volume. Ciò permette una misura più precisa.

Il volume delle perdite è causato principalmente dal movimento trasversale del filtro ma anche dalle perdite per trafileamento che avvengono nel circuito di aspirazione.

La lunghezza effettiva è standardizzata dalla normativa ISO 10054 come 405 mm, in base alla misura nominale degli smoke meter standard. Tale valore è ottenuto da un volume di misura di 330 cm^3 e da un area del filtro di 8.15 cm^2 .

Inoltre occorre considerare che il valore di 405 mm è calcolato per gas a 298 K di temperatura e ad 1 bar di pressione.

Il volume del gas all'interno del dispositivo viene aspirato tramite un sistema di pompaggio: nei modelli precedenti avveniva tramite una pompa alternativa, mentre per i nuovi 415 e 415s avviene con una pompa a membrana continua.

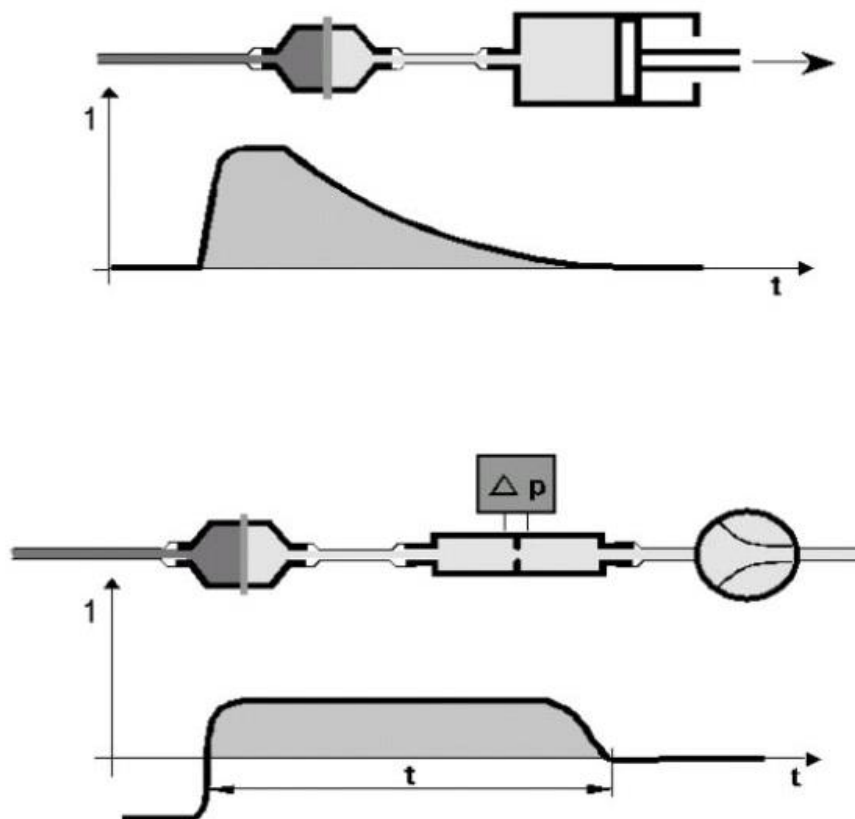


Figura 28 Grafico relativo al volume aspirato nel tempo

Tramite quest'ultima è possibile effettuare misure variando sia il volume di prova (che prima era dato dalla dimensione del pistone alternativo) che il tempo.

In figura si evidenziano le differenze tra il vecchio sistema (primo diagramma) e nuovo: l'immissione di gas è più costante nel tempo, permettendo così al filtro di assorbire la fuliggine con una maggior precisione. Questo è utile soprattutto durante misure molto brevi o con pochi inquinanti presenti nelle quali con il precedente sistema non si poteva ottenere un risultato accettabile. La sequenza del test prevede innanzitutto una misura della portata massica di gas di scarico in [kg/sec] che attraversa il filtro secondo la formula:

$$\dot{m} = k \cdot \sqrt{\Delta p \cdot \frac{p}{T}}$$

Dove k: è un fattore di proporzionalità;

Δp : delta di pressione al orifizio d'ingresso

p: pressione assoluta dell'ambiente di prova

T: temperatura

Successivamente l'algoritmo è:

- Calcolo della lunghezza effettiva in base ai volumi;
- Acquisizione dell'annerimento del filtro tramite un rifrattometro;
- Determinazione della concentrazione di fuliggine in base all'annerimento e alla lunghezza effettiva;
- Invio degli indici calcolati all'utente.

Tutte queste operazioni sono schematizzate nell'immagine; in funzione del tempo vengono riportate le aperture delle valvole a solenoide, l'attivazione della pompa a diaframma, del sensore di misura e lo scorrimento del filtro di carta.

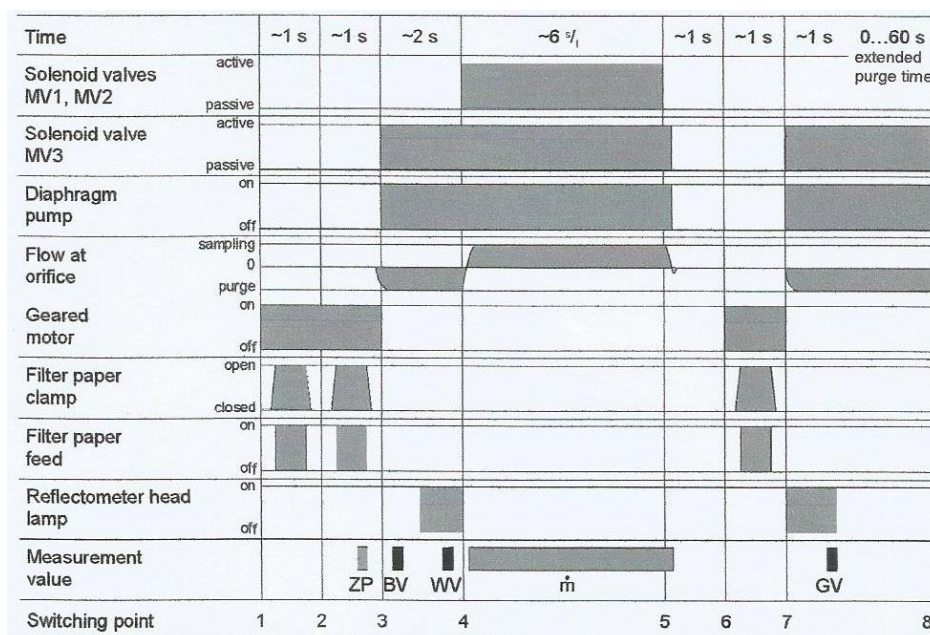


Figura 29 Sequenza di misura

4.1.2 – Parametri di Misura

L'annerimento del filtro viene misurato tramite un sistema ottico composto da:

1. Rifrattometro;
2. Sorgente luminosa;
3. Fotodiiodo;
4. Filtro pulito;
5. Filtro annerito.

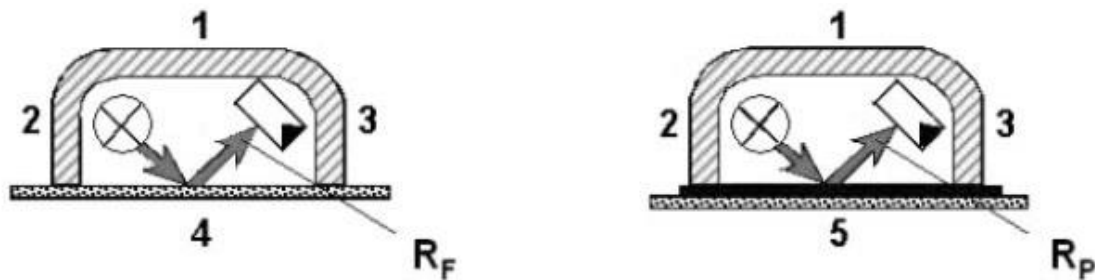


Figura 30 Sensore ottico per l'acquisizione del Paper Blackening

Un led invia un segnale luminoso che viene riflesso dal filtro su un fotodiiodo che ne calcola l'intensità. Quando il filtro è completamente bianco (R_f) avremo un segnale forte che cala all'aumentare dell'annerimento del filtro (R_p). L'intensità viene valutata in Volt.

Definiamo quindi l'annerimento (PB, **paper blackening**) come:

$$PB = \frac{100 - R_R}{10}$$

Dove R_R è la luminosità relativa (**relative radiance factor**) ottenibile dai sensori:

$$R_R = \frac{R_P}{R_F} \cdot 100\%$$

Il PB avrà valori da 0 a 10 secondo la scala definita da BOSCH. Nelle versioni precedenti (AVL 409) i valori erano calcolati da 0 a 9 e quindi venivano riadattati.

Il PB è quindi il principale dato di output; esso viene interpretato diversamente a seconda delle normative:

- ISO 10054: utilizzata in Europa ed America;
- JIS D 8004: utilizzata in Asia.

Secondo la normativa occidentale, se la misura avviene in condizioni standard (lunghezza effettiva di 405 mm, $T=298$ K e $p=1$ bar) definiamo il “**Filter Smoke Number**”:

$$FSN = PB$$

Questa misura è relativa, infatti dipende dal fattore R_R .

Secondo la normativa orientale invece la misura dell'annerimento è assoluta. Definiamo quindi il “**Pollution Level**”:

$$PL = 100 - 1,15 \cdot R_A[\%]$$

La corrispondente lunghezza effettiva è circa 380 mm.

Il parametro R_A viene definito luminosità assoluta (**Absolute radiance factor**) come:

$$R_A = \frac{R_P}{R_S} \cdot 100$$

Dove R_P : è come prima il valore rilevato;

R_S : è il valore standard del bianco (ad esempio dell'ossido di magnesio)

Il PL ci fornisce quindi un valore percentuale di inquinamento del filtro. Di norma la macchina viene calibrata con $R_A=43,5\%$ da cui deriva $PL=50\%$; può accadere di trovare misure negative (-2% fino a -10%) per filtri completamente puliti.

Infine definiamo l'offset come valore di PL in % per un filtro pulito.

A partire dall'offset otteniamo il PL equivalente a partire dall'FSN con la seguente formula:

$$PL_{equi}[\%] = FSN \cdot \frac{100 - offset}{10} + offset$$

Di seguito è riportata graficamente la relazione fra PL e FSN con un offset di -6%.

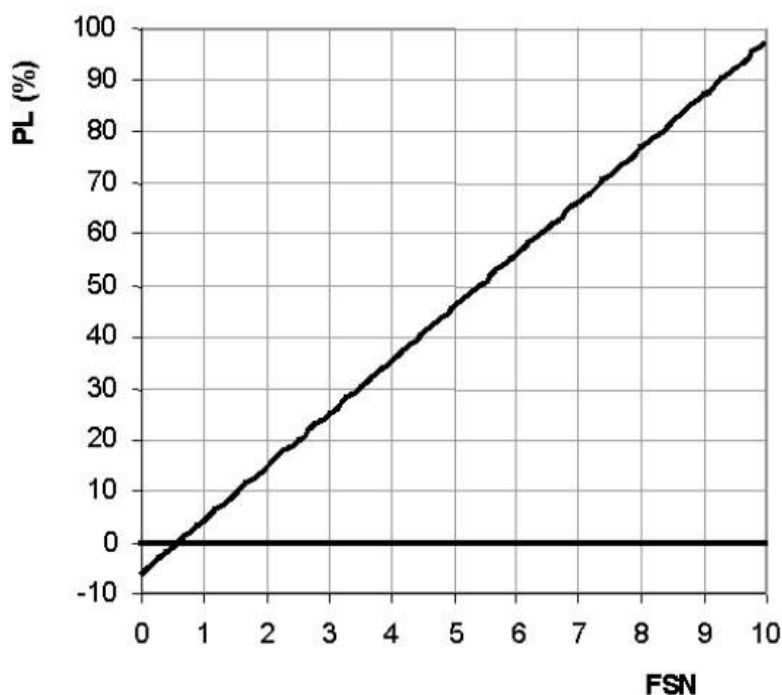


Figura 31 Relazione tra PL (%) e FSN

4.1.3 – *Fattori di Influenza*

Il **PB** misurato può essere influenzato da diversi fattori:

- Volume di misura e tempo;
- Carta del filtro;
- Umidità e temperatura.

Il volume viene scelto in modo che il PB risultante sia compreso fra 0.5 e 8; in particolare i valori raccomandati da AVL vanno da 1 a 5.

Considerando l'indice FSN (ricordando che è uguale al PB in condizioni standard) otteniamo 1 con una prova da 1 litro. Per i valori limite avremo una riduzione di 1/3 per prove ad alto contenuto di fuliggine (FSN>6) e l'incremento di 2,5 volte per misure a basso contenuto (FSN<1), in modo da permettere al filtro di annerirsi.

Al volume è legato anche il tempo di misura dalla relazione:

$$6 \text{ secondi} \leftrightarrow 1 \text{ litro}$$

Per tempi lunghi di misura occorre scaldare il device (*Heating Option*) così da evitare fenomeni di condensa.

La carta filtro è composta da fibre di cotone molto difficili da produrre e sottoposte a test molto severi; AVL fornisce la standard **S&S tipo 597LA** da conservare a una temperatura compresa fra 15 e 28° C ed un'umidità tra il 30 e il 60%.

Infine temperatura e umidità sono legate e possono portare a variazioni anche significative. Di seguito sono riportati schemi relativi alla variazione di FSN durante due prove svolte in cui varia il tempo di campionamento e la temperatura del *device*.

Nella prima abbiamo un FSN di partenza di 0.8:

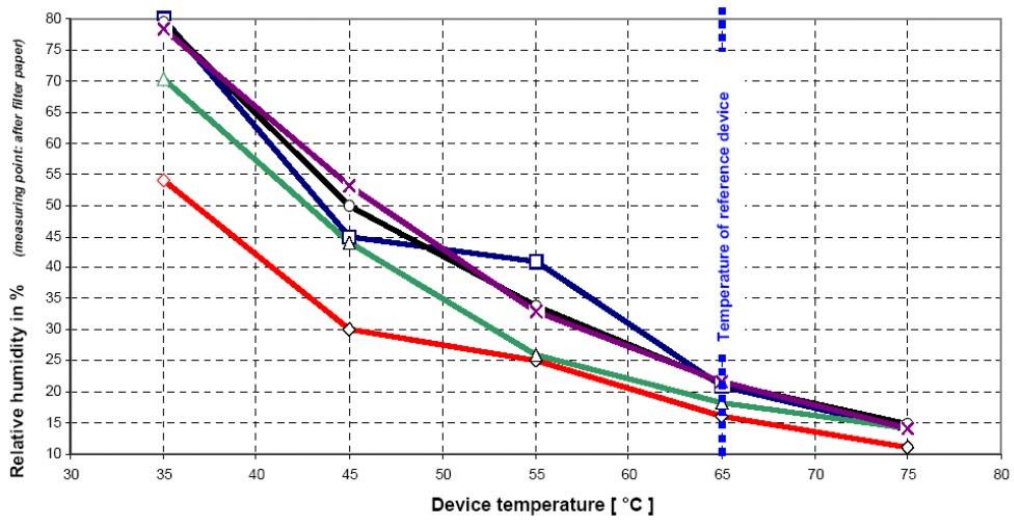


Figura 32 Diagramma Umidità relativa-Temperatura Device con FSN=0.8

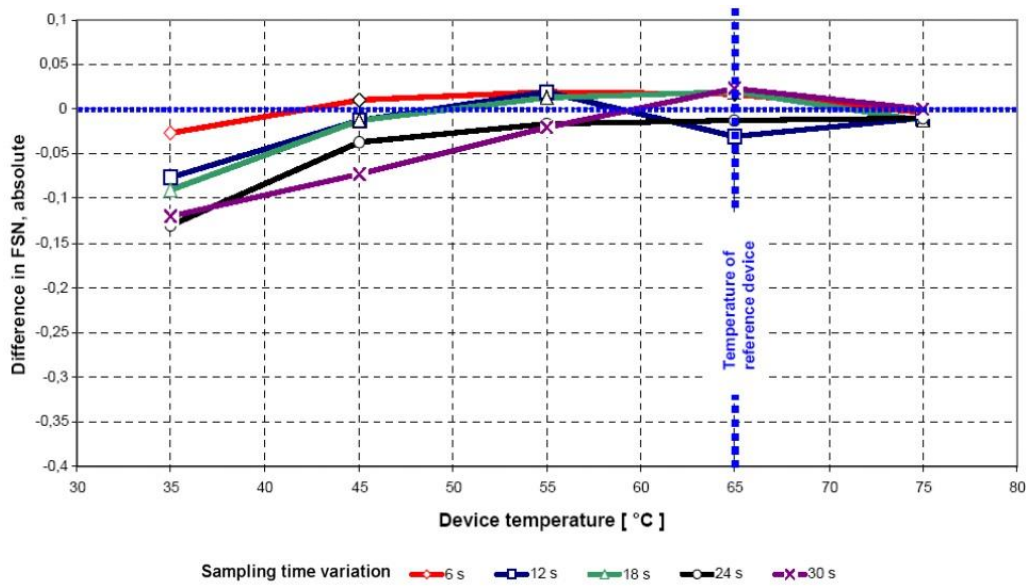


Figura 33 Diagramma FSN Differenziale-Temperatura Device con FSN=0.8

Nella seconda abbiamo FSN di 2:

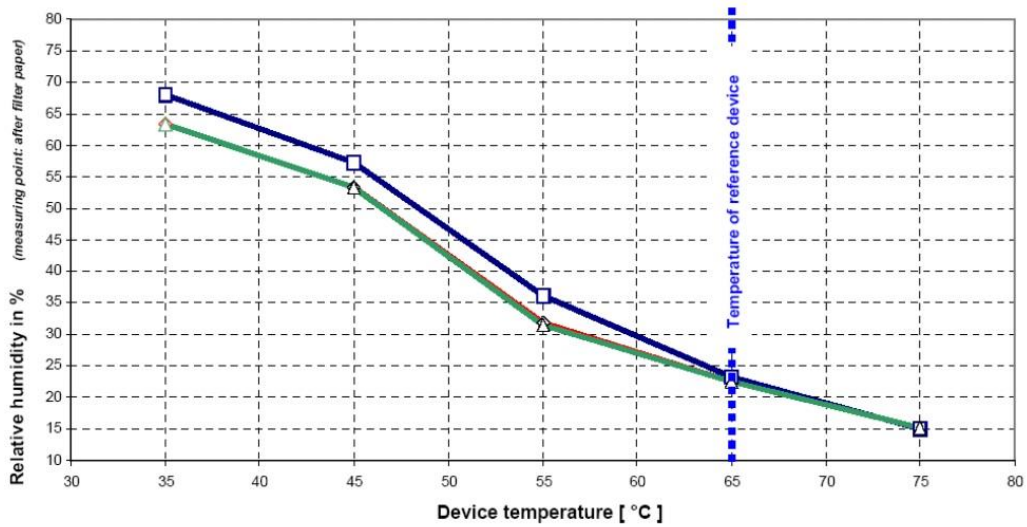


Figura 34 Diagramma Umidità relativa-Temperatura Device con FSN=2

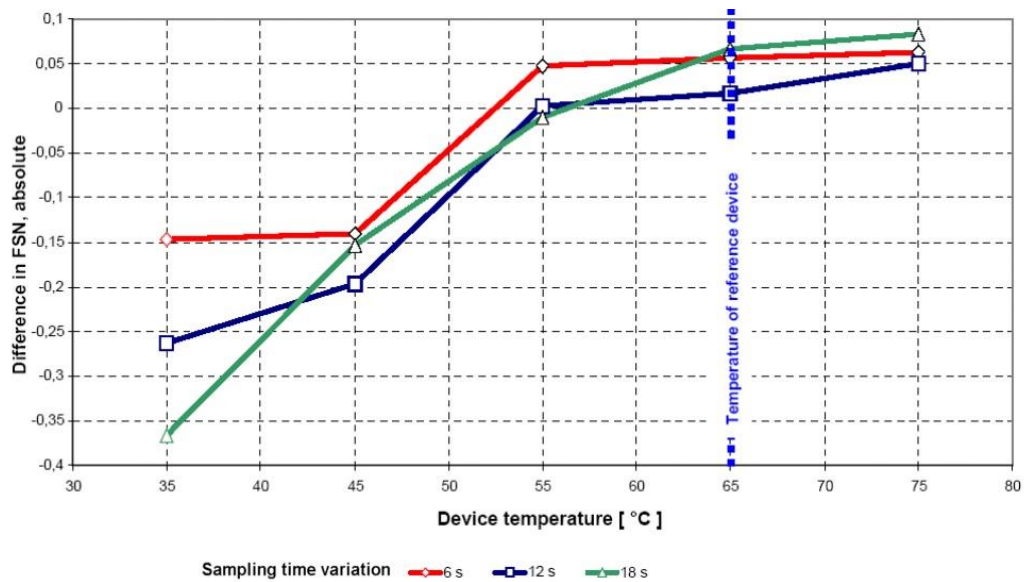


Figura 35 Diagramma FSN Differenziale-Temperatura Device con FSN=2

All'aumentare dell'indice FSN raddoppia l'umidità assoluta nei gas di scarico: quindi con l'aumentare della temperatura e per tempi lunghi di misura è facile la formazione di acqua in dispositivi non riscaldati.

Nelle condizioni standard la condensazione dell'acqua avviene per temperature inferiori a 55° C, quindi mantenere il dispositivo ad un riferimento di 65° C ci consente di evitare errori dovuti alla presenza di liquidi.

4.2 – VI AVL_415s

Lo smoke meter, come la bilancia, comunica con un linguaggio seriale utilizzando il protocollo AK. Il 415s è collegato, tramite una porta X4 EXTERN posta sul dispositivo, ad un test bed (AVL DITEST CDS) che svolge la diagnosi chimica dei gas con il supporto dell'AVL DITEST GAS 1000. Questo insieme di device (opzionali rispetto al dispositivo standard) non è integrabile tramite un sistema di VI.

Il programma della tesi quindi si limita a consentire le operazioni di base dello smoke meter:

- Misure singole o multiple dei coefficienti P_B , FSN e PL;
- Misure con un volume prefissato di gas di scarico;
- Misure con una durata prefissata della prova.

Il collegamento con il pc HOST avviene tramite la porta COM1 posta sul 415s.

4.2.1 – Segnali di Input e Output

Andiamo a definire i segnali di input ed output che andremo poi ad integrare nel VI.

In uscita avremo i dati che concernono lo stato del device (connessione, stato, errore, ...) e tutti i risultati dei test da eseguire.

I parametri di **stato** sono:

- *Device info*: come per la bilancia indica il nome dell'apparecchio a cui si è connessi con la sigla **415S Vx.xx**; questo output serve anche come ulteriore verifica dello stato della connessione: se la casella non è riempita correttamente c'è sicuramente un errore.
- *Acknowledge Message*: è l'unione delle caselle Error e Action del VI AVL_733s; nel riquadro durante la comunicazione appare sia la decodifica dell'operazione in atto, sia l'eventuale errore inviato dal dispositivo.
- *Action*: è un insieme di output a LED che indicano lo stato della bilancia; l'operazione di misura richiede un ciclo che dura diversi secondi in cui la macchina non riceve altri input; grazie ad *action* ci viene indicato se la bilancia è pronta ad eseguire misure, sta calcolando o c'è un errore.

I **risultati** dei test invece sono:

- *Sampling Volume*: indica il volume della colonna di gas su cui si svolge la misura;

- *Effective sampling Length*: fornisce la lunghezza effettiva riferita al volume aspirato;
- *Paper Blackening (P_B)*, FSN, PL: sono gli indici definiti dalla normativa;
- *Soot Concentration*: otteniamo la percentuale di fuliggine presente sul filtro.

In ingresso al dispositivo, similmente alla bilancia, avremo sia comandi utili all'utilizzo sia i parametri di setup del dispositivo.

I **controlli** sono:

- *Set Ready*: interrompe tutte le azioni in atto e prepara il device al campionamento;
- *Purge*: come spiegato in precedenza il volume nocivo e la condensa provocano misure sbagliate; grazie a questo comando si attua una depurazione, con aria pulita, del condotto di misura per cercare di ridurre entrambe al minimo;
- *Paper Economy Mode*: attiva la modalità di utilizzo con la quale viene utilizzata la minor porzione di filtro possibile;
- *Reset*: come per la bilancia riporta il device allo stato iniziale con tutti i parametri standard;
- *Query*: è il comando che avvia le prove; a seconda del codice funzione che inseriamo otteniamo risultati differenti.

I **parametri** impostabili sono relativi al setup del dispositivo durante le misure:

- *Number of Samples*: imposta il numero di prove da fare consecutivamente;
- *Sampling Volume*: fissa il volume di gas da aspirare durante il test;
- *FSN Mean Value*: abilità oltre alle misure standard anche il calcolo dell'indice FSN medio, nel caso di misure multiple.

Questi sono i principali segnali di comunicazione tra cella e pc. Nell'applicazione non verranno inseriti tutti i comandi relativi alla calibrazione e al sistema interno.

4.2.2 – Interfaccia grafica

L'interfaccia, similmente alla bilancia, è costituita da due schede per rendere semplice e schematico l'utilizzo.

Nella prima, chiamata Setup, saranno presenti tutti i controlli e gli indicatori relativi allo stato della macchina e ai parametri di misura.

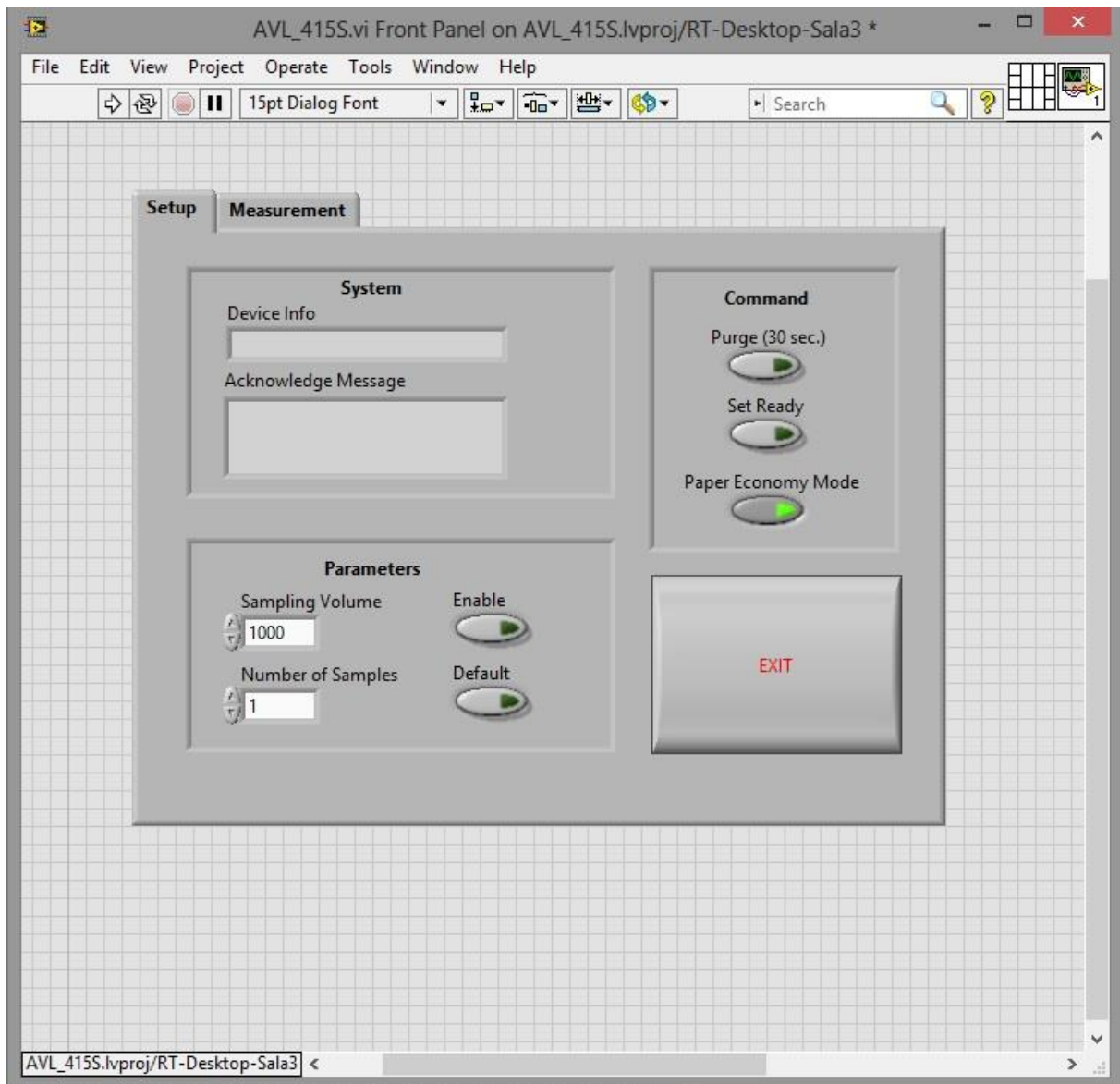


Figura 36 Scheda Setup del pannello frontale

La scheda è divisa in tre riquadri: *System*, *Parameters* e *Command*; in aggiunta è presente il solito comando **EXIT** utile a disconnettere il device e chiudere l'applicazione.

Il blocco *System* contiene gli indicatori *Device Info* e *Acknowledge Message* già spiegati. In *Parameters*, con l'utilizzo di controlli a numerici digitali, si possono variare i dati relativi al volume di misura e al numero di campionamenti da eseguire. I parametri vengono inviati al dispositivo solo quando si spinge il bottone **Enable**; azionando **Default** invece si inviano alla bilancia i valori standard forniti da AVL.

Infine nel riquadro *Command* sono presenti i controlli relativi alle funzioni **Purge**, **Set Ready** e **Paper Economy Mode**. I comandi sono in forma di pulsanti on/off ma con reazioni differenti:

- *Purge*: una volta attivato rimane su on fino alla fine del ciclo (circa 30 sec.); terminata la depurazione il pulsante torna in automatico su off;
- *Set Ready*: il pulsante ha un'azione meccanica detta *Latche when pressed*, ossia una volta spinto torna subito su off mentre esegue l'ordine;
- *Paper Economy Mode*: il controllo rimane su on fintanto che non viene rispinto il pulsante; a quel punto si sposta su off e disattiva la modalità di risparmio.

La scheda *Measurement* invece contiene tutti i controlli e gli indicatori relativi alle prove.

Nell'immagine possiamo vedere la sua struttura:

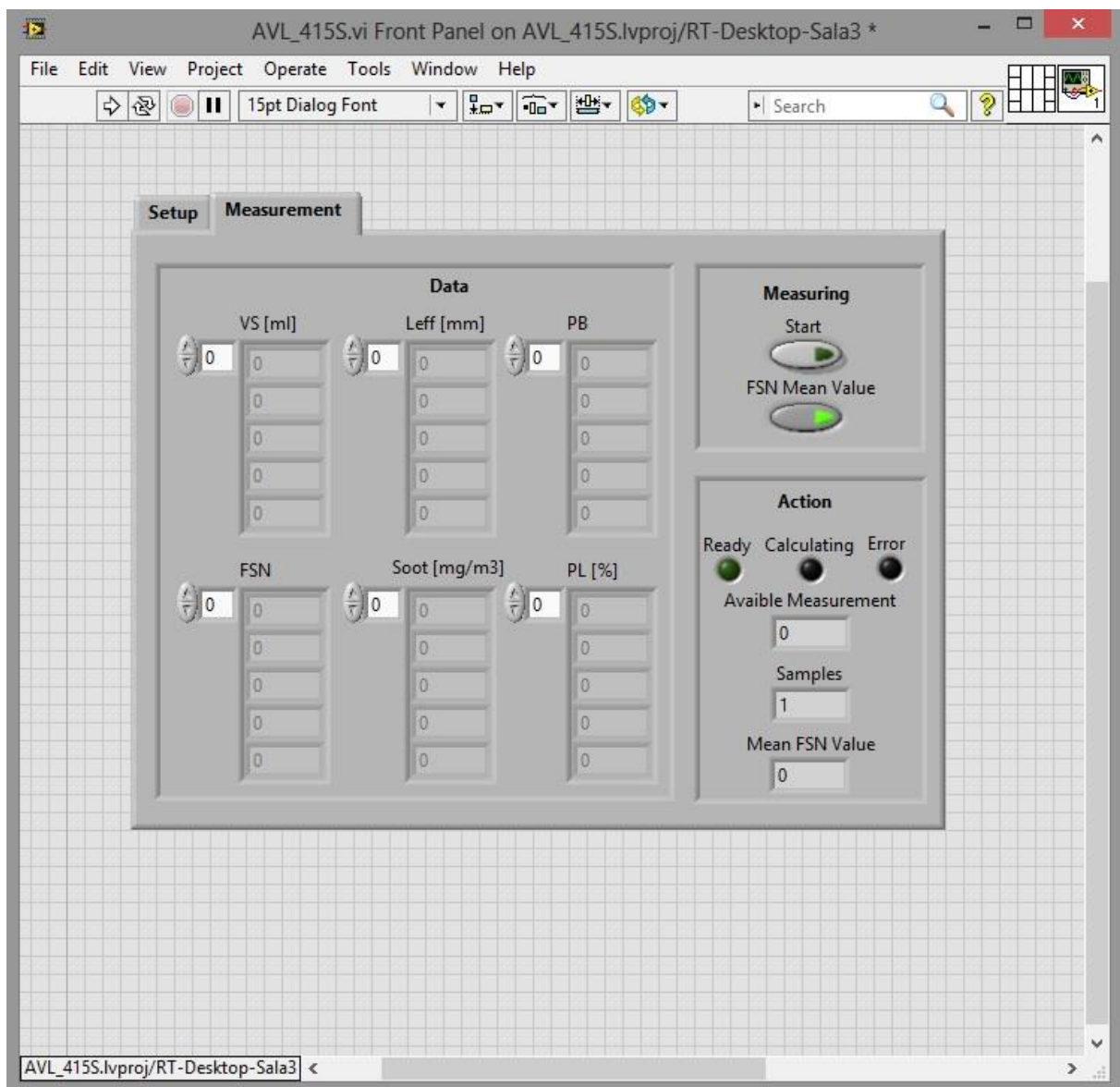


Figura 37 Scheda *Measurement* del pannello frontale

Anche questa scheda è divisa in riquadri utili a semplificare l'utilizzo.

Nel blocco **Data** sono presenti tutti i risultati dei test; gli indicatori sono in forma vettoriale nel caso in cui si svolgano misure multiple. Ogni colonna contiene un dato del quale sono specificate anche le unità di misura.

Il riquadro **Measuring** invece contiene i due comandi di misura:

- Star: invia al device il comando di Query iniziando il test;
- FSN Mean Value: come già specificato attiva la modalità di calcolo dell'indice medio su prove multiple.

Infine **Action** è il blocco di indicatori che indica l'operatività dello smoke meter. Tramite tre LED di colore differente (verde, giallo e rosso) ci viene indicato rispettivamente se la macchina è pronta ad eseguire misure (Ready), se sta ancora eseguendo un test (Calculating) o se è presente un errore (Error), la cui codifica è presente nell'indicatore apposito della scheda Setup.

Come già detto non sono presenti comandi sulla calibrazione, operazione delicata da eseguire manualmente, o sul sistema interno, come consigliato da AVL. Tutto quello che concerne invece il contenuto della fuliggine (soot) non è integrabile, ma può essere ottenuto dal test bed fornito Dall'interfaccia possiamo solo conoscere la % presente sul filtro.

4.2.3 - AK-Protocol

Il protocollo utilizzato per la comunicazione è lo stesso della bilancia; nonostante alcune differenze sul codice funzione, la modalità master-slave rimane invariata.

I messaggi si dividono in:

- Command message: è l'insieme di tutto ciò che viene inviato dal pc HOST verso il device, analogamente alla bilancia;
- Acknowledge Message: sono le risposte che invia lo smoke meter all'utente.

La composizione dei messaggi è:

Command Message	
Byte	Function
1	<STX>
2	Ignored
3 ... 6	Function code SXXX ... control commands EXXX ... setting commands AXXX ... query commands
7	Blank
8	K
9	Channel number (always 0)
.	Data (variable)
.	can also be omitted (depending on function code)
.	
n th byte	<ETX>
Acknowledgement Message	
Byte	Function
1	<STX>
2	Ignored
3 ... 6	Function code (same as command)
7	Blank
8	Error status 0 no error 1 ... 9 error (counted upwards in cycles)
.	Data (variable)
.	can also be omitted (depending on function code)
.	
n th byte	<ETX>

Figura 38 Schema di composizione dei messaggi

Come si può notare la struttura è molto simile al caso del 733s.

La modalità di suddivisione dei codici funzione è:

- Sxxx per i controlli relativi ai comandi da imporre al device;
- Exxx per impostare i setup;
- Axxx per qualunque tipo di query da richiedere.

La differenza dalla bilancia avviene nei codici funzione. Di seguito sono riportati solo i codici utilizzati nel VI; l'elenco completo è disponibile sul manuale operativo.

I comandi di controllo (**Sxxx**) permettono di svolgere le principali azioni:

- **SRES**: reset, reimposta il dispositivo nello stesso stato dell'accensione;
- **SREM**: porta lo smoke meter in Modalità Remota; il comando è sempre disponibile;
- **SRDY**: annulla e interrompe le procedure in corso e prepara il dispositivo allo stato utile ai test;
- **SMES**: inizia la misura oppure continua con quella in atto; se il dispositivo è in stato Ready il test si avvia immediatamente;
- **SPUL**: invia al device il comando di depurazione (*Purge*) per la durata massima di 30 secondi;
- **SPSE**: avvia la modalità *Paper Economy* con la quale si utilizza la minor quantità di filtro possibile;
- **SPSA**: disattiva la modalità *Paper Economy*;

I principali comandi d'interrogazione (**Axxx**) servono a richiedere i dati delle nostre misure:

- **AKEN**: identifica il dispositivo con 415S Vx.xx;
- **ASTZ**: indica sia lo stato del dispositivo, sia se sta eseguendo le seguenti azioni:
 1. *Remote*;
 2. *Purge*;
 3. *Paper Economy Mode*;
 4. *SMES*;
- **APAP**: richiede allo smoke meter quante prove sono disponibili controllando la quantità di filtro rimasto. Il numero di prove è arrotondato a 50 quando la modalità di risparmio è disattivata;
- **AMZY**: è la funzione responsabile dei dati nella scheda Setup; nel blocco delle variabili ci vengono forniti:

1. *sampling volume* in ml; il range di misura è [50 ... 20000]; impostando 0 si attiva l'Auto-Range con il quale il sistema si autoregola;
 2. *number of samples*: il numero di campionamenti nell'intervallo [1 ... 5];
- **AMWE**: comunica gli indici acquisiti durante la prova:
 1. *sampling volume* a 25 °C e 1 bar in ml con una precisione di 1 decimale;
 2. *effective sampling length* in mm con una precisione di 1 decimale;
 3. *paper blackening* [0 ... 10] con una precisione di 2 decimali;
 4. filter smoke number (FSN) nello stesso intervallo con una precisione di 2 decimali;
 5. *soot concentration* in mg/m³ con una precisione di 2 decimali;
 6. *pollution level (PL)* in % con una precisione di 1 decimale;
 - **AFSN**: fornisce l'indice FSN in misure multiple e quindi il valore medio; nel VI utilizziamo solo la funzione che ci consente di ricevere FSN Mean Value.

Infine i comandi d'impostazione (**Exxx**) servono a personalizzare i parametri di sistema e misura:

- **EMZY**: indica al dispositivo i parametri di misura riferiti a:
 1. *sampling volume* nel range [50 ... 20000] o 0 per l'Auto-Range;
 2. il numero di prove nel range [1 ... 5].

Definiti i codici funzione si procede con lo schema a blocchi del programma.

4.2.4 – AVL_415S

Il VI dell'interfaccia grafica è **AVL_415S** programmato all'interno di un progetto **RT** per consentire il solito dialogo continuo. Il loop temporale viene eseguito con una frequenza di 1 kHz. Di seguito è riportato lo schema a blocchi:

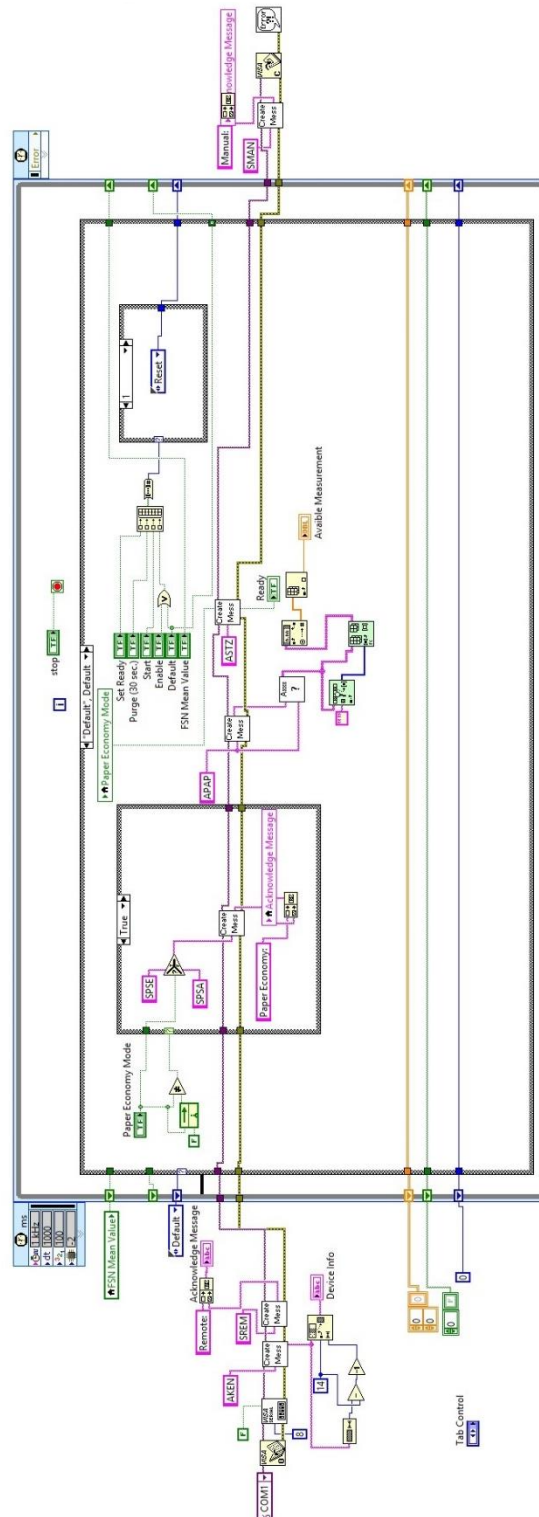


Figura 39 Schema a blocchi del caso Default

All'avvio l'applicazione svolge una sequenza di azioni (VISA Serial, AKEN e SREM) tali da permettere il collegamento con il device e da fornire le informazioni iniziali delle schede *Setup (Device Info)* e *Measurment (Ready)* presenti nell'interfaccia.

Lo schema, come per la bilancia, è programmato tramite una struttura *case* in cui gli input forniti fungono da trigger per spostarsi dal caso di **Default**. I casi della struttura sono:

- *Default*;
- *Reset*;
- *Purge*;
- *Query*;
- **SMES**;
- *Result*;
- *Parameters*;
- **FSN**.

Il caso **Default**, come per la bilancia, è il responsabile della comunicazione in continuo con il dispositivo. Al suo interno avviene sia l'invio degli input che le interrogazioni riguardo lo stato (quantità di prove disponibili, modalità *Paper Economy, Ready*).

A differenza del VI AVL_733s, in cui ogni caso svolgeva una funzione per poi riportarsi a **Default**, in questo VI i case relativi alle misure (*Query, SMES, Result*) sono **sequenziali**; questo espediente è dovuto alla durata delle misure: infatti, non essendo misure istantanee, necessitano di una temporizzazione differente per non creare conflitti con il **timed loop** del progetto RT. Viene quindi brevemente analizzata la struttura dei suddetti per rendere chiaro il funzionamento del VI.

Si accede al case **Query** spingendo sul pannello frontale il bottone Start. Al suo interno tramite l'interrogazione **AMZY** si chiede allo smoke meter quanti campionamenti sono stati fissati. Una volta ottenuta la risposta si separa il messaggio tramite il codice funzione e si seleziona solo il contenuto che interessa. La stringa ottenuta viene poi trasformata in un numero intero inviato sia all'indicatore *Number of Samples* del pannello sia ad un blocchetto chiamato *Initialize Array* con il quale sono inizializzati n vettori di dimensione 6 (gli elementi sono gli output VS, Leff, PB, FSN, soot e PL) a seconda del numero di prove.

In parallelo viene anche creata con la stessa funzione una costante 0 (indice della prova) e un vettore di n costanti logiche impostate su falso. La funzione di questo vettore viene spiegata nei paragrafi successivi. A questo punto ci spostiamo al caso successivo; di seguito si può osservare lo schema a blocchi di Query:

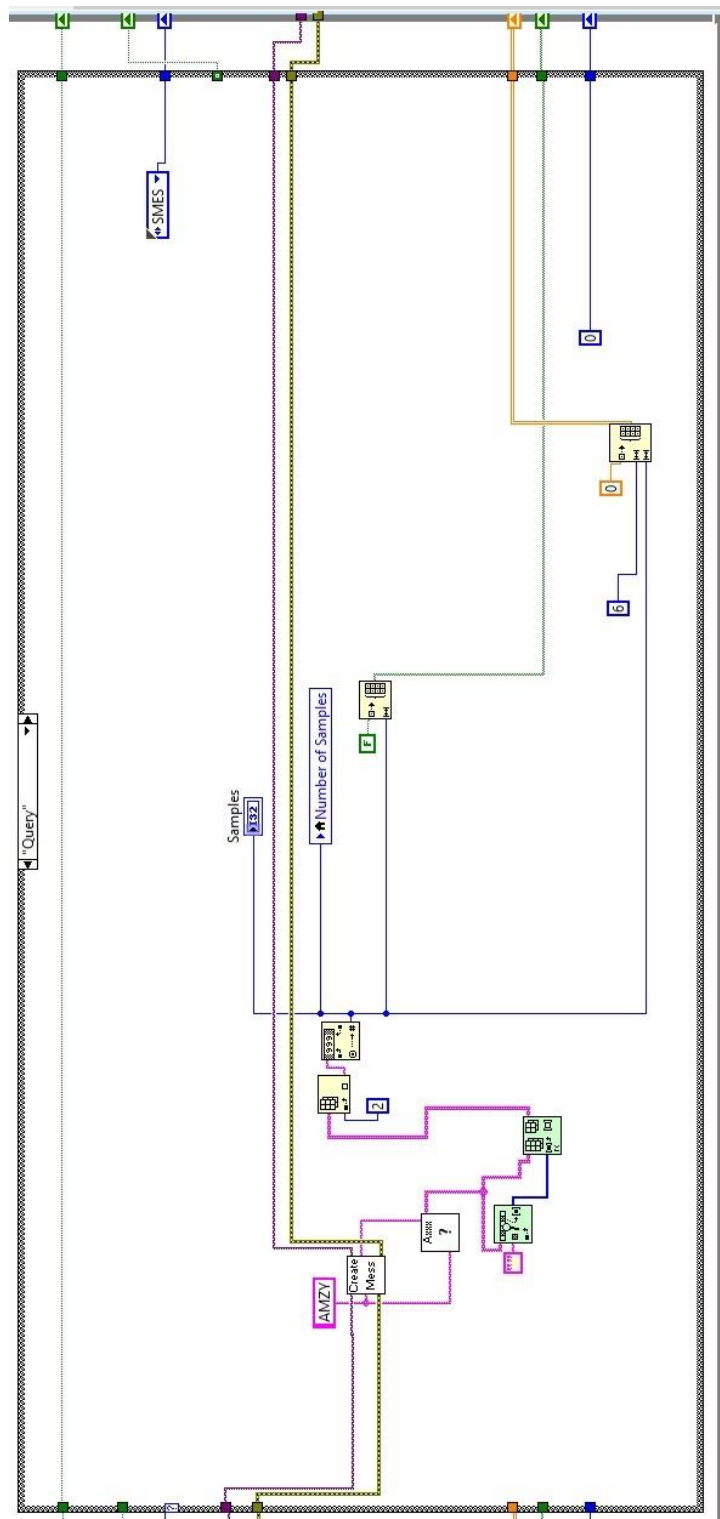


Figura 40 Schema a blocchi del caso Query

Il caso successivo è **SMES**: serve semplicemente ad inviare il comando che imposta la macchina allo stato Ready; nel *Acknowledge Message* del pannello frontale comparirà il messaggio *Measuring* e si illuminerà il LED verde relativo.

Svolte queste semplici azioni si prosegue con il caso seguente.

4.2.5 – *SubVI*

Anche in questo VI sono presenti i SubVI utili alla comunicazione con il device; non ci sono particolari differenze con quelli della bilancia a livello logico se non qualche modifica nei messaggi di errore o nelle uscite relative agli stati (Paper Economy, Ready, ...) non presenti nel 733s.

Per questo motivo vengono elencati di seguito, ma senza la spiegazione del funzionamento:

- **avl_astz_SMES**: chiede lo stato del device e delle sue funzioni principali;
- **avl_create_message**: crea i messaggi da inviare e ne codifica gli errori nelle risposte;
- **avl_create_parameter**: fornisce al dispositivo i parametri che vogliamo cambiare;
- **interrogation**: crea la stringa contenente i Data Block delle risposte.

5 – MDrive 23 Plus

5.1 - Dispositivo

L'ultimo dispositivo integrato nel sistema è un motore passo-passo lineare **MDrive 23 Plus** acquistato dalla **Schneider Electric**.



Figura 43 MDrive 23 Plus

Il device è equipaggiato con un motore ibrido bipolare **NEMA 23** che compie ad ogni step una rotazione di 1.8° e con un sistema di controllo elettronico integrato. La rotazione dello stepper viene trasformata in movimento lineare tramite la vite costruita in acciaio inox non magnetico.

Il motore è costituito da uno statore, che consiste nel classico insieme di avvolgimenti a 8 poli, e da un rotore formato da ruote dentate affiancate, solidali all'albero di rotazione, e costituite da un nucleo magnetico.

La corrente eccita in successione i poli dello statore secondo il senso in cui vogliamo ruotare la vite.

Nell'immagine si può vedere come il passaggio di eccitazione dal polo 1 al polo 2 causa la rotazione della ruota dentata di un fattore costante; nel caso del NEMA 23 di 1.8°

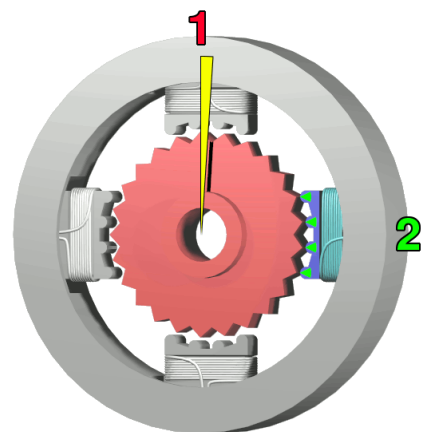


Figura 44 Funzionamento del motore NEMA 23

I vantaggi di questo dispositivo sono:

- la possibilità di compiere piccole rotazioni in ambedue le direzioni;
- bloccaggio in determinate posizioni;

- l'azionamento può essere controllato dal computer con grande precisione;
- elevata robustezza meccanica e scarse inerzie; anche se il movimento a scatti prevede una velocità massima di 1500 rpm e la generazione di vibrazioni per il movimento a scatti.

5.2 VI R_NEMA_Main

All'interno della sala, MDrive viene utilizzato per movimentare la farfalla del motore al banco. La comunicazione con l'utente avviene attraverso un cavo **Ethernet**, utilizzando un protocollo **TCP**.

Il programma della tesi si pone come obiettivo sia il comando della rotazione da remoto che la calibrazione del dispositivo, così da renderlo indipendente dalla sua disposizione in sala rispetto al motore.

5.2.1 I segnali di Input e Output

Al contrario di quanto avvenuto con i dispositivi AVL, per questo *device* la richiesta di movimento dello stepper e la risposta sono confrontati passo passo per verificare l'effettiva attuazione del comando; infatti, durante il funzionamento, avremo come input la posizione desiderata (obj, objective) e come risposta dal software avremo quella ottenuta (act, actual). Allo stesso modo tutti i parametri da impostare per il movimento sono suddivisi in act e obj, poiché possono essere cambiati anche durante il funzionamento.

Solo la calibrazione, che di norma si svolge al montaggio del motore al banco, è utilizzata come input tradizionale e come output fornisce i valori act della posizione della vite e l'escursione massima eseguibile.

Andiamo quindi a definire i segnali (input/output) che vengono scambiati da sistema e sala:

- *NEMA 23 IP*: fornisce al sistema l'indirizzo IP del dispositivo;
- *No Connection*: indica che la connessione tra dispositivo e pc è saltata;
- *Not Calibrated/Calibration*: sono rispettivamente l'output che indica la non calibrazione del dispositivo e l'input che avvia la procedura di calibrazione;
- *Throttle Obj/Act*: indica in percentuale l'apertura della farfalla desiderata e quella attuale raggiunta dal sistema;
- Parametri di movimento e posizione: svolta la calibrazione il sistema fornisce tutti i dati relativi tra cui corrente di utilizzo, posizione rispetto allo 0, velocità di movimento e accelerazione.

Noti i segnali è stata costituita l'interfaccia grafica.

5.2.2 Interfaccia grafica

Come i VI precedenti anche **R_NEMA_Main** è stata suddivisa in schede tali da rendere schematico e semplice l'utilizzo.

La prima è *Control*, tramite la quale si svolge l'attività di movimentazione.

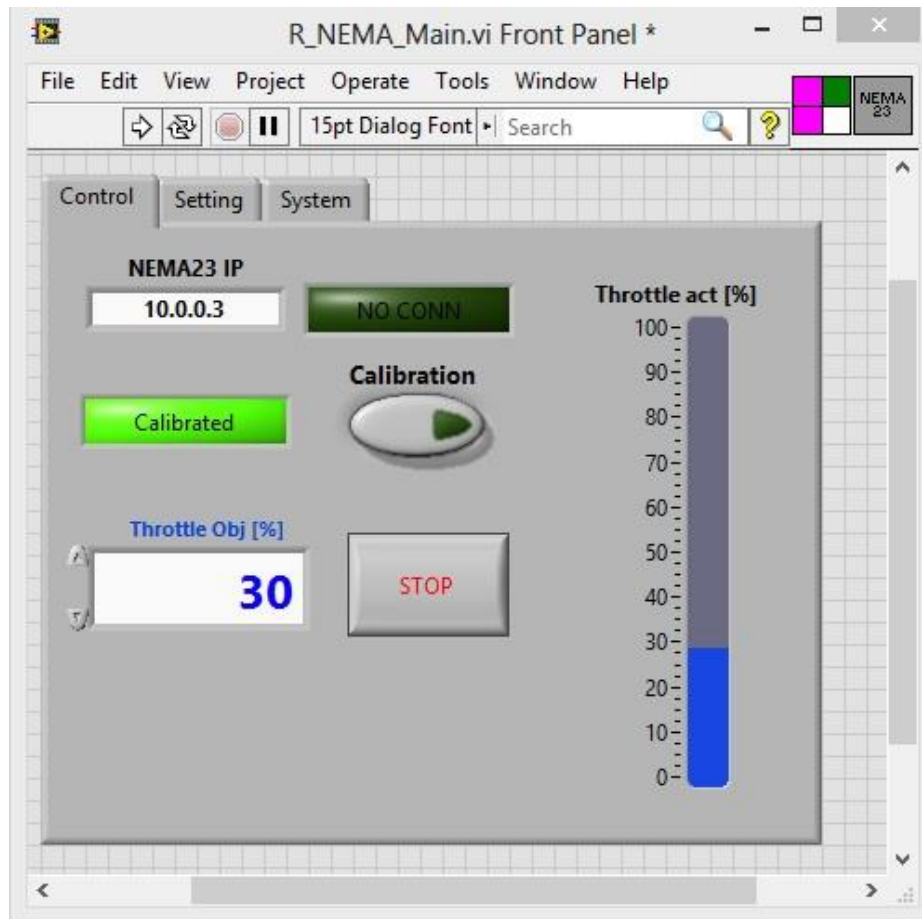


Figura 45 Scheda Control del pannello frontale

Nella parte alta sono presenti tutti i dati relativi allo stato del *device*, quindi connessione e calibrazione. Quando il LED verde diventa rosso, compare la scritta “*Not Calibrated*”; tramite il bottone relativo si può eseguire quindi la calibrazione.

Nella parte bassa, oltre al solito **STOP** che disconnette ed interrompe l'applicazione, è presente il comando con cui si procede a movimentare il dispositivo; considerando l'utilizzo su farfalla viene indicato con *Throttle* con una scala percentuale che indica l'apertura, dalla totale (100%) alla minima (0%, anche se in realtà una parte di portata passa).

Nell'indicatore a barra di destra è invece rappresentata l'effettiva apertura della valvola riscontrata dal sistema. Per cambi notevoli (0-100%) può risultare un ritardo tra *obj* e *act*.

La seconda scheda è invece relativa all'impostazione del dispositivo.

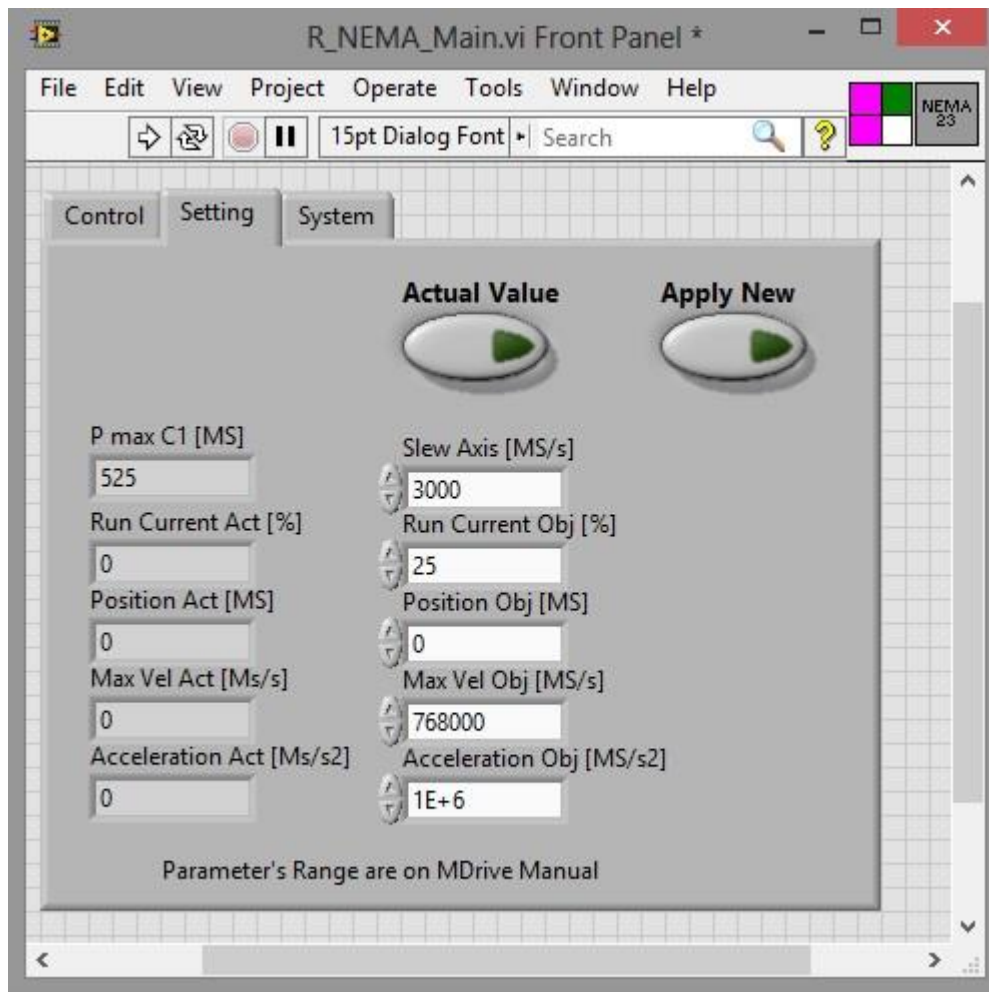


Figura 46 Scheda Setting del pannello frontale

I due controlli in alto servono rispettivamente a richiedere i **setting** in uso e ad applicare le eventuali modifiche.

Le due colonne in basso invece rappresentano i parametri su cui è possibile agire (sempre nell'accezione act a sinistra e obj a destra):

- *Run Current*: indica la corrente utilizzata dal motore in percentuale;
- *Position*: fornisce la posizione del counter in **Microstep** (MS); di default il valore è 0;
- *Max Velocity*: indica la velocità massima con cui sono eseguiti gli spostamenti in microstep al secondo [MS/sec];
- *Acceleration*: modifica l'accelerazione dello stepper in MS/sec².

Inoltre è presente **Slew Axis** e **P max C1** che verranno spiegati successivamente nel paragrafo dedicato alla calibrazione.

Infine l'ultima scheda è riservata al sistema ed indica solamente gli errori e la frequenza di aggiornamento del programma.

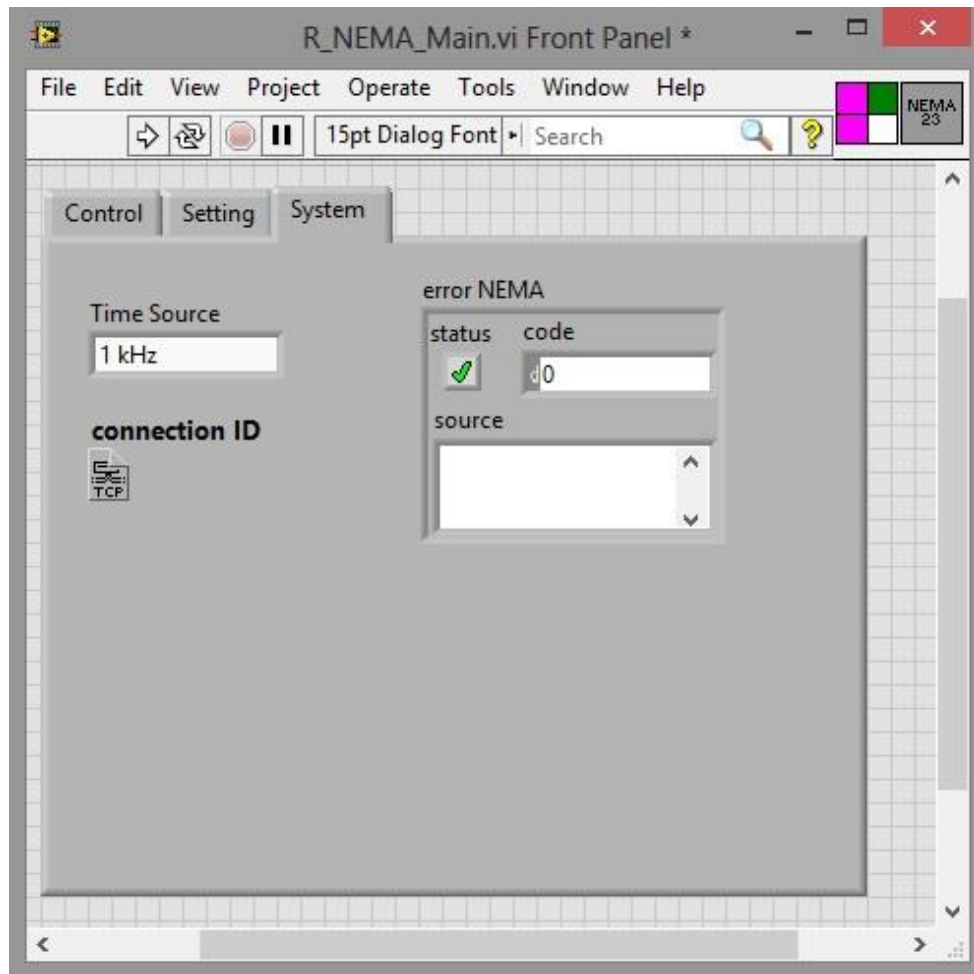


Figura 47 Scheda System del pannello frontale

5.2.3 MCode

La comunicazione avviene tramite cavo Ethernet e trasmissione TCP; il linguaggio utilizzato dal device invece è **MCode**: è costituito, come il seriale da stringhe, ma la composizione del messaggio si limita semplicemente ad un codice funzione (di uno o due caratteri) seguito dal carattere di invio.

Di seguito sono riportati in ordine alfabetico tutte le funzioni utilizzate all'interno del VI:

- **VA**: crea una variabile per l'utente; nel caso della tesi si chiamerà Q3;
- **MA** (*Move to Absolute Position*): muove lo stepper ad una posizione assoluta del sistema di riferimento calibrato;

- **RC** (*Run Current*): indica la corrente con cui il motore agisce in percentuale; di default è settata su 25%;
- **P** (*Position Counter*): indica la posizione dello stepper in Microstep;
- **VM** (*Maximum Velocity*): fornisce la velocità massima con cui il motore si deve spostare; il massimo è di 5000000 MS/S quando è disabilita la modalità di encoding e di 200000 quando è attiva; i default sono 768000 e 30720;
- **A** (*Acceleration*): fornisce l'accelerazione massima del motore in MS/sec²;
- **EE** (*Enable Encoding*): assume i valori 1 e 0 che rispettivamente abilita o disabilita la modalità di encoding che determina la posizione assoluta;
- **SM** (*Stall Detected Mode*): assume i valori 1 e 0 che rispettivamente abilita o disabilita la modalità di stallo;
- **MR** (*Move to a Relative Position*): muove lo stepper ad un posizione relativa rispetto alla corrente;
- **SL** (*Slew Axis*): indica la velocità di movimento degli assi;
- **MV** (*Moving*): ordina il movimento dello stepper.

5.2.4 Main e subVI

Il **Main** è costituito dalla solita *case structure* che consente di spostarsi da un caso all'altro in funzione di determinati input. La struttura è composta da quattro stati:

- *Inizialization*;
- *Move*;
- *Parameters*;
- *Setting*.

Inizialization serve a connettere il dispositivo e a verificarne la calibrazione. Nella pagina successiva è mostrato lo schema a blocchi.

Nella parte sinistra è possibile osservare la presenza di un blocco relativo alla comunicazione TCP-IP, con la verifica di eventuale errore tipica del protocollo. Se tale verifica ha esito positivo si illumina il LED relativo sul pannello, con la scritta "*NEMA Conn.*" che indica l'avvenuta connessione.

Nella parte destra invece viene verificata la calibrazione richiedendo al sistema l'entità della variabile **Q3**: tale dato è programmabile all'utente e in questa applicazione viene utilizzato per definire la posizione degli assi di riferimento. Quindi se la lunghezza della stringa Q3 è nulla, cioè il dato non esiste, si illumina nel pannello il LED di non calibrazione del *device*.

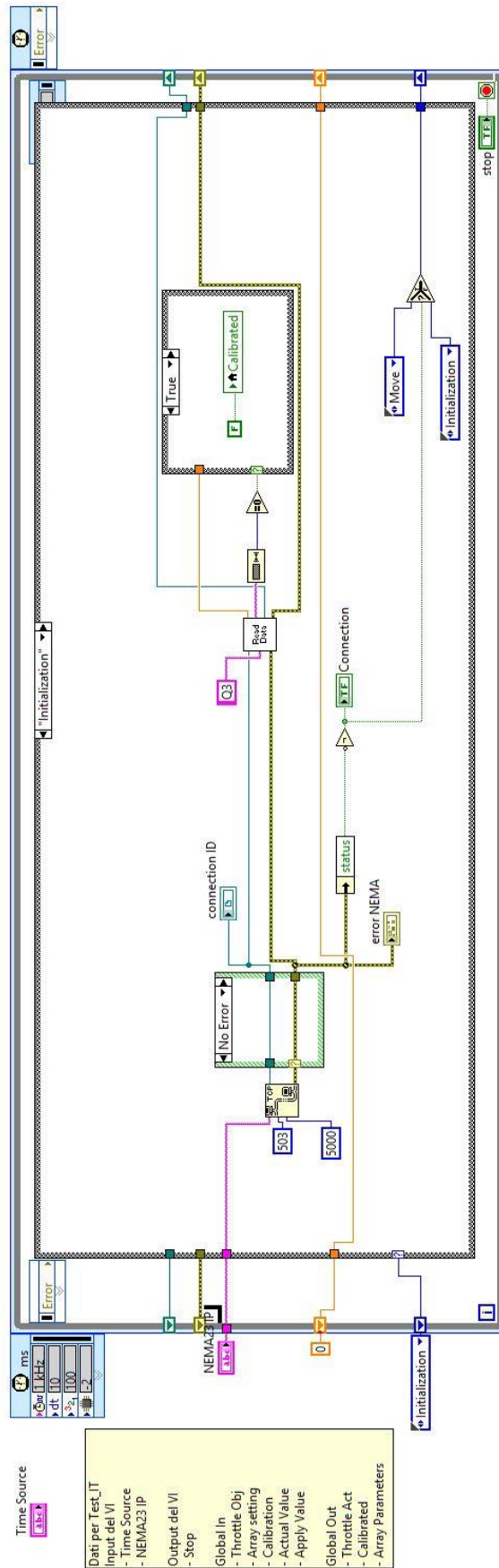


Figura 48 Schema a blocchi del caso Inizialization

Eseguita la connessione, il programma si sposta al case successivo, **Move**, che sarà quello di *Default*.

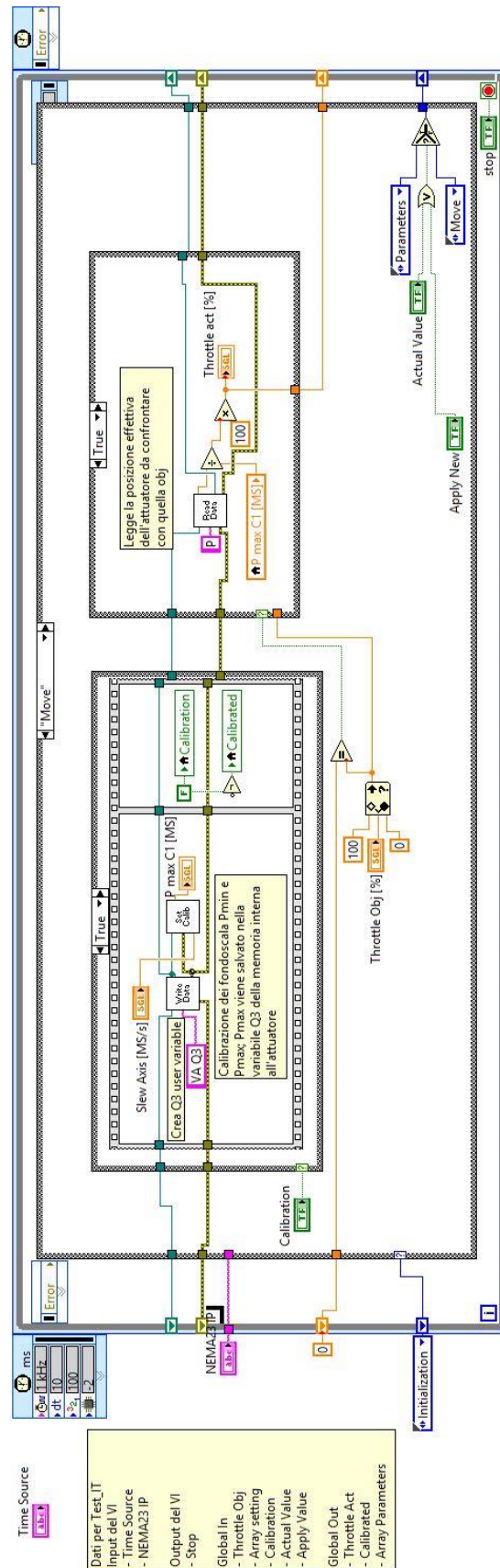


Figura 49 Schema a blocchi del caso Move

Nella parte di sinistra si procede con la calibrazione se il rispettivo comando sul pannello Control viene premuto; il processo di calibrazione sarà descritto in seguito.

Nella parte destra si legge il valore di *Throttle Obj* richiesto dall'utente e lo si confronta con quello presente nel dispositivo (inizialmente è 0 di default). Dal confronto si ottiene risposta affermativa o negativa il cui responso conduce ai seguenti casi:

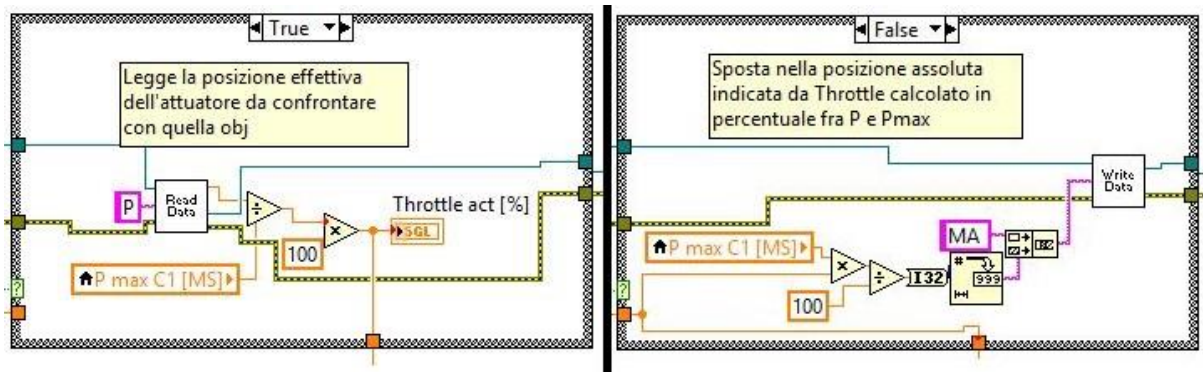


Figura 50 Case structure conseguente alla verifica della posizione

Se la risposta è positiva significa che la farfalla è nella posizione giusta; quindi viene aggiornato il valore di *Throttle act* con la posizione corrente. Al contrario, se la risposta è negativa e il target non cambia, si procede con il movimento (funzione **MA**) della quantità indicata da *Throttle Obj*: occorre quindi trasformare la percentuale in movimento assoluto moltiplicandola prima per l'escursione massima *P max* e poi per 100.

Nel caso in cui i controlli **Apply New** e **Actual Value** siano entrambi falsi si rimane all'interno del caso *Move*, altrimenti il programma avanza al case **Parameters**. Di questo caso non è riportato lo schema a blocchi in quanto ci si limita a leggere i valori correnti dei parametri settati dal *device*. Alla fine dell'interrogazione *Actual Value* torna al valore False, disattivando il controllo. Se invece si è spinto *Apply New*, invece di ritornare a *Move*, si procede all'ultimo caso presente nel programma, ossia **Setting**. Nella pagina successiva è presente lo schema a blocchi.

Innanzitutto il programma confronta i valori *Act* con quelli *Obj*, dopodiché scrive al *device* i soli valori cambiati rispetto ai correnti. Una volta inviati i dati il controllo *Apply New* torna al valore False e il programma si riporta al case *Move*.

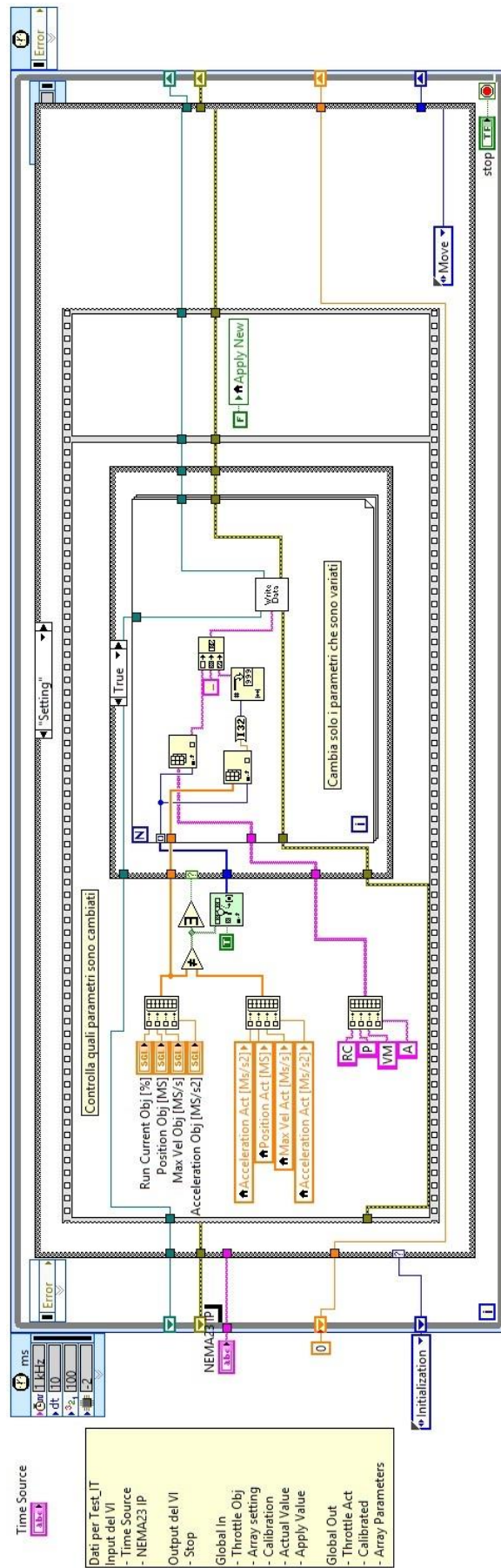


Figura 51 Schema a blocchi del caso Setting

All'interno del programma sono presenti tre subVI:

- **Write Data:** invia semplicemente il comando al device tramite TCP;
- **Read Data:** legge la risposta del dispositivo;
- **Set Calib:** si occupa del processo di calibrazione.

Se i primi due sono semplicissimi (e per questo non viene descritto lo schema a blocchi), l'ultimo è complicato quanto importante. Per questo viene riportata di seguito una breve descrizione e nella pagina successiva lo schema a blocchi.

Setcalib è una successione di operazioni necessarie alla calibrazione. Segue quindi la descrizione puntuale delle varie sequenze (solo quelle dove si compiono azioni, non quelle dedicate alla temporizzazione):

1. Dopo aver abilitato la modalità di Encoder (EE=1, per avere posizioni assolute calcolate dall'encoder interno), occorre disattivare la modalità di stallo (SM=0) per procedere;
2. Le sequenze successive servono a preparare la calibrazione: si sposta dalla posizione corrente il motorino di 6000 MS e si imposta la corrente del motore al 15%;
3. Inizia la calibrazione vera e propria: si muovono gli assi con escursione negativa (SL – il dato che il sistema legge in **Slew Speed**) fino a che lo stepper incontra un carico resistente eccessivo; ciò porta la macchina allo stallo registrando la posizione corrente come fondo scala minimo.
4. Si procede allo stesso modo ma nel verso opposto (SL +) per ottenere l'escursione massima. Tale posizione diventa **P max C2**;
5. Vengono reimpostate la corrente di default (25%) e si disabilita la modalità di encoder (EE=1);
6. A questo punto si richiede al *device* la posizione che diventerà **P max C1** utilizzata dal programma e memorizzata in **Q3**.
7. Infine si riporta lo stepper al valore assoluto 0 del nuovo sistema di riferimento.

Queste operazioni descrivono il ciclo di calibrazione e concludono il capitolo sul MDrive Plus.

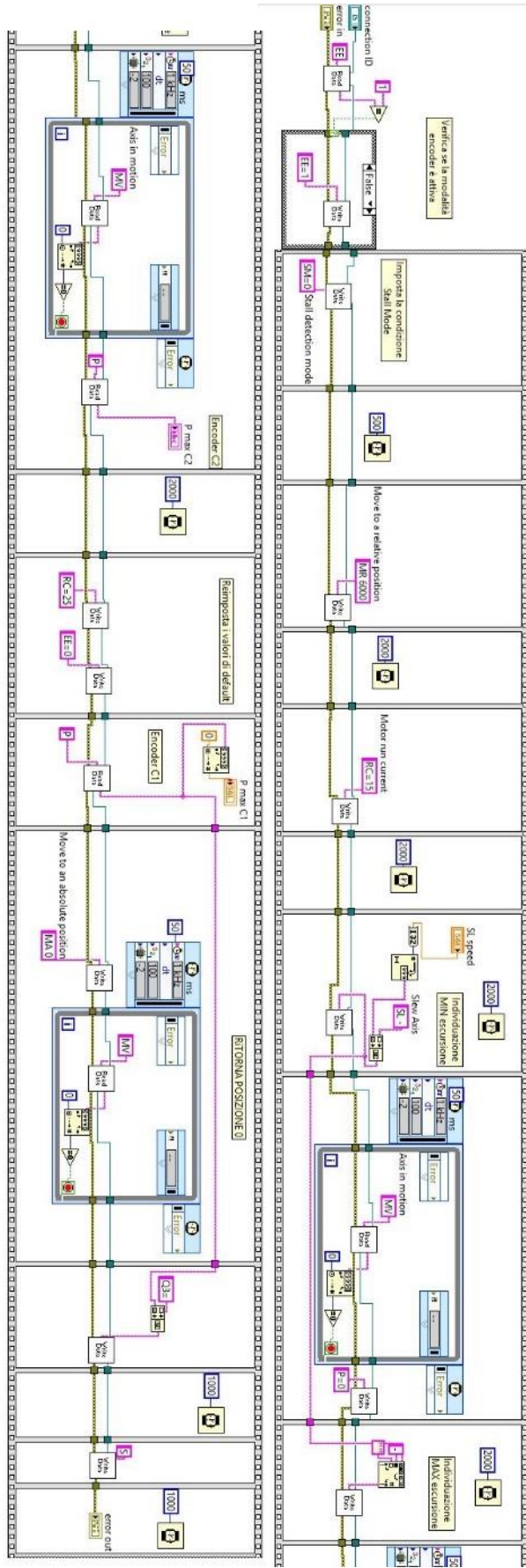


Figura 52 Schema a blocchi del SubVI Setcalib

6 – Implementazione in TestIT

TestIT, come già spiegato nell'introduzione, è il software responsabile della gestione di tutti i dispositivi presenti in sala. In questo capitolo definiamo le operazioni effettuate per implementare i VI **AVL_733s**, **AVL_415s** e **Mot_Lin** all'interno del programma.

6.1 Gestione dei Dati

La struttura di TestIT prevede due programmi principali: **rMain** e **Host**. Il primo opera sul pc RT della cella, mentre il secondo sul pc dell'utente. Tra i due avviene uno scambio continuo di dati regolato dal pc Real-Time e fruibile tramite Host. Tale flusso di dati necessita di alcune caratteristiche fondamentali:

- Protocollo di comunicazione: è il mezzo con cui avviene lo scambio tra un pc e l'altro;
- Creazione di variabili globali per rendere fruibili i dati a più applicazioni;
- Metodo di transito dei dati.

Di seguito abbiamo un veloce descrizione di questi tre punti.

6.1.1 – Protocollo di Comunicazione

Come noto il computer Real-Time gestisce tutti i dati in ingresso ed uscita alla sala, che comunica all'Host con un protocollo UDP.

User Datagram Protocol (UDP) è un protocollo di rete utilizzato per l'invio di bit di dati sotto forma di pacchetti, in particolare per applicazioni "leggere" e per la trasmissione real-time. Insieme a **Transmission Control Protocol (TCP)**, è il protocollo più comunemente usato; la differenza fra i due si verifica nel controllo degli errori che avviene in TCP, mentre manca in UDP con conseguente perdita di affidabilità ma incremento di velocità.

In una trasmissione TCP l'algoritmo è il seguente:

- Invio dei pacchetti da un IP all'altro;
- Verifica da parte del protocollo della ricezione dei pacchetti da parte del destinatario;

- Invio da parte del destinatario della conferma dell'avvenuta ricezione; se il mittente non riceve la conferma rinvia il pacchetto fino a che non riceve risposta; questa azione continua per un certo periodo dopo il quale si considera il destinatario offline.

Durante ogni invio e ricezione il pacchetto viene controllato per accertarsi che sia esente da errori.

Nel protocollo UDP invece i pacchetti sono inviati di continuo senza alcuna verifica sugli errori né sulla ricezione: capiamo subito che questo comporta da un lato una scarsa affidabilità con la possibilità di perdita dei dati, dall'altro un aumento notevole di velocità grazie all'assenza di latenza dovuta a controlli e riorganizzazione.

Nel caso di TestIT, per il timed loop interno al programma, la comunicazione avviene con dati che rimangono costanti per tutto il ciclo di **10 ms**; grazie a questa costanza possiamo permetterci di perdere qualche dato, essendo in ogni caso sicuri che la comunicazione avvenga all'interno del loop. Quindi optiamo nella quasi totalità degli scambi (es. l'attuatore utilizza la TCP) per il protocollo UDP che risulta più veloce e più leggero a livello computazionale.

6.1.2 – R_gl

In programmazione esistono variabili locali, utilizzabili all'interno del programma in cui vengono descritte, e **variabili globali** che sono definite esternamente e pertanto condivisibili da più VI. Per rendere possibile quindi il dialogo tra rMain e Host occorre che tutti i dati siano definiti da globali. All'interno del progetto quindi esiste un particolare VI con la funzione di creare al suo interno tutte le variabili globali che si muovono all'interno di TestIT così da renderle immediatamente rintracciabili e fruibili da tutti i programmi del progetto: questo VI è **R_gl**. Esso non è altro che un magazzino in cui rMain e Host stoccano le informazioni elaborate in attesa di essere inviate tramite UDP.

Nell'immagine è possibile vedere **R_gl**; nei cerchi rossi sono presenti gli indicatori di input e output che rappresentano le variabili globali delle nostre applicazioni.

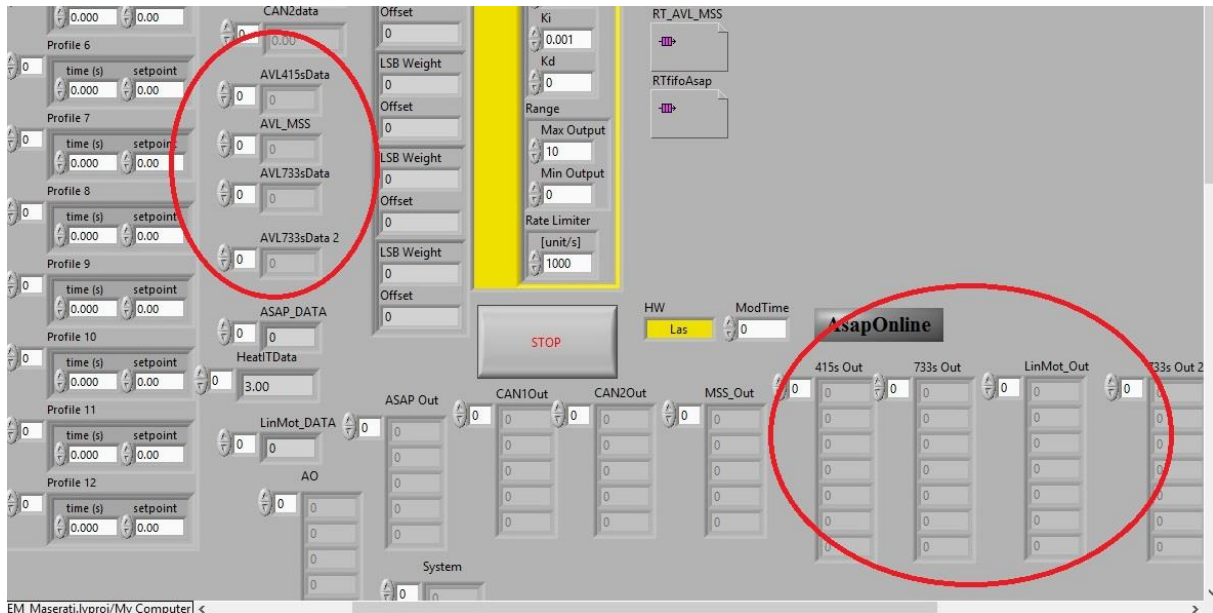


Figura 53 VI R_gl

6.1.3 – FIFO

Il termine **FIFO** è l'acronimo inglese di First in First Out ed identifica il metodo di transito con cui compiliamo le nostre variabili globali. In programmazione viene utilizzato per creare buffer di dati; non prevede inizializzazione, ma il semplice impacchettamento dei dati.

6.2 – Struttura dei SubVI

Le interfacce create per l'utilizzo dei device AVL vanno trasformate in **SubVI** così da poter essere richiamate all'interno di TesIT; la caratteristica principale dei sottoprogrammi è il **riquadro dei connettori**, tramite il quale possono essere utilizzati come black box a cui inviare dati e da cui riceverli elaborati. Per costruirlo si crea in ogni subVI un array che contiene tutti gli input e uno che contiene tutti gli output, dopodiché si sceglie il pattern con cui richiamare l'applicazione. Gli array appena generati verranno poi inviati a R_gl sotto forma di variabili globali consentendo tutto il processo di comunicazione descritto sopra.

6.2.1 – AVL 733s

Nell'immagine è riportato lo schema a blocchi del subVI della bilancia:

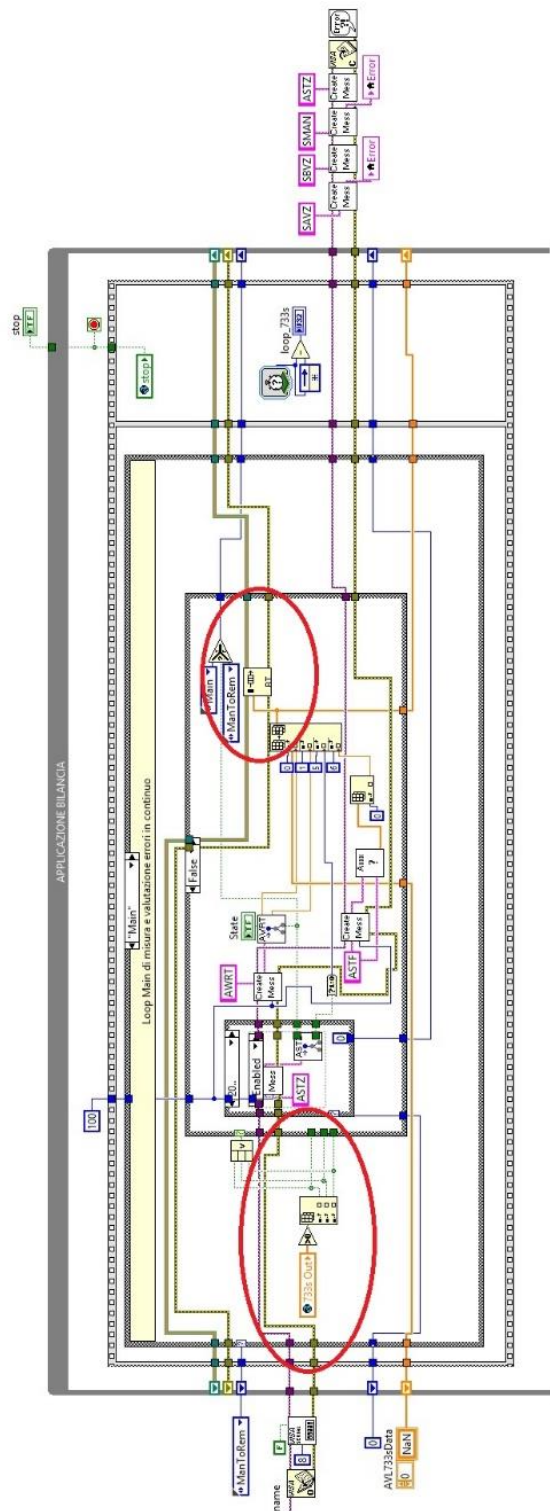


Figura 54 Schema a blocchi del SubVI AVL_733S

Nel cerchio di sinistra è possibile osservare l'array di variabili globali **733sOut** che in uscita dall'Host fornisce i comandi alla bilancia: sono i tre input logici riferiti a *Reset*, *Refill*, *Start/Stop* ed un ulteriore double per la quantificazione dell'intervallo di misura. Nel cerchio di destra invece è presente la creazione dell'array che riscriverà la variabile globale di input all'Host; tale vettore contiene i segnali di *Filling level*, *Consumption*, *Consumption Start/Stop*, *Measuring Time*, *Filling* ed *Error Code*.

La struttura a timed loop è stata sostituita da una *sequence*, che garantisce solo il corretto ordine della operazioni, poiché il programma è già all'interno del loop temporale di rMain.

6.2.2 AVL 415s

Analogamente alla bilancia, il programma va cambiato affinché sia presente un solo array di input ed uno solo di output. Lo schema a blocchi del subVi è:

Nel cerchio rosso è riportata la funzione che raccoglie l'array contenente gli output dello smoke meter per scriverli all'interno della variabile globale che verrà inviata all'Host.

Come input invece si ha un array con i seguenti segnali logici: Ready, Purge, Start, Enable, Default e FSN Mean che svolgono le stesse funzioni precedenti.

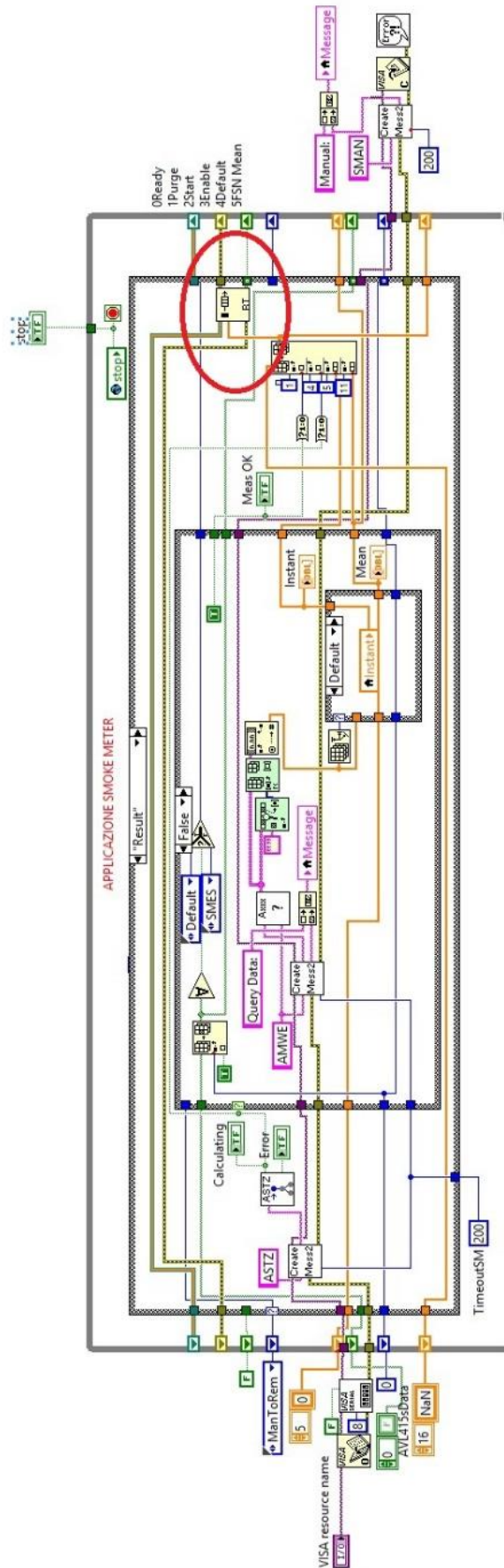


Figura 55 Schema a blocchi del SubVI AVL_415S

6.2.3 – NEMA23

Infine si riporta il programma dell'**attuatore lineare**. Anche in questo caso all'interno dei cerchi rossi sono presenti i termini con cui generiamo la comunicazione: a sinistra si nota la TCP che comunica le istruzioni al motorino, mentre a destra è presente l'output inviato tramite variabile all'Host.

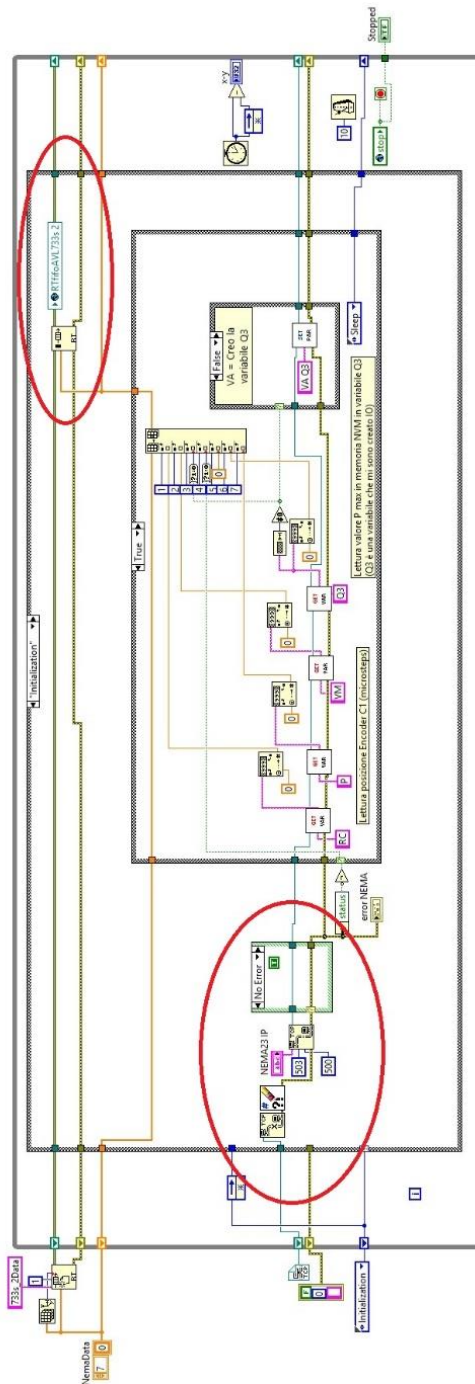


Figura 56 Schema a blocchi del SubVI NEMA23

6.3 – rMain

Una volta create, le variabili globali di ogni subVI sono richiamate all'interno di rMain dove vengono impacchettate in un buffer per essere spedite all'Host.

Nella figura sottostante è rappresentata la struttura che aggiorna le variabili globali e le invia tramite UDP.

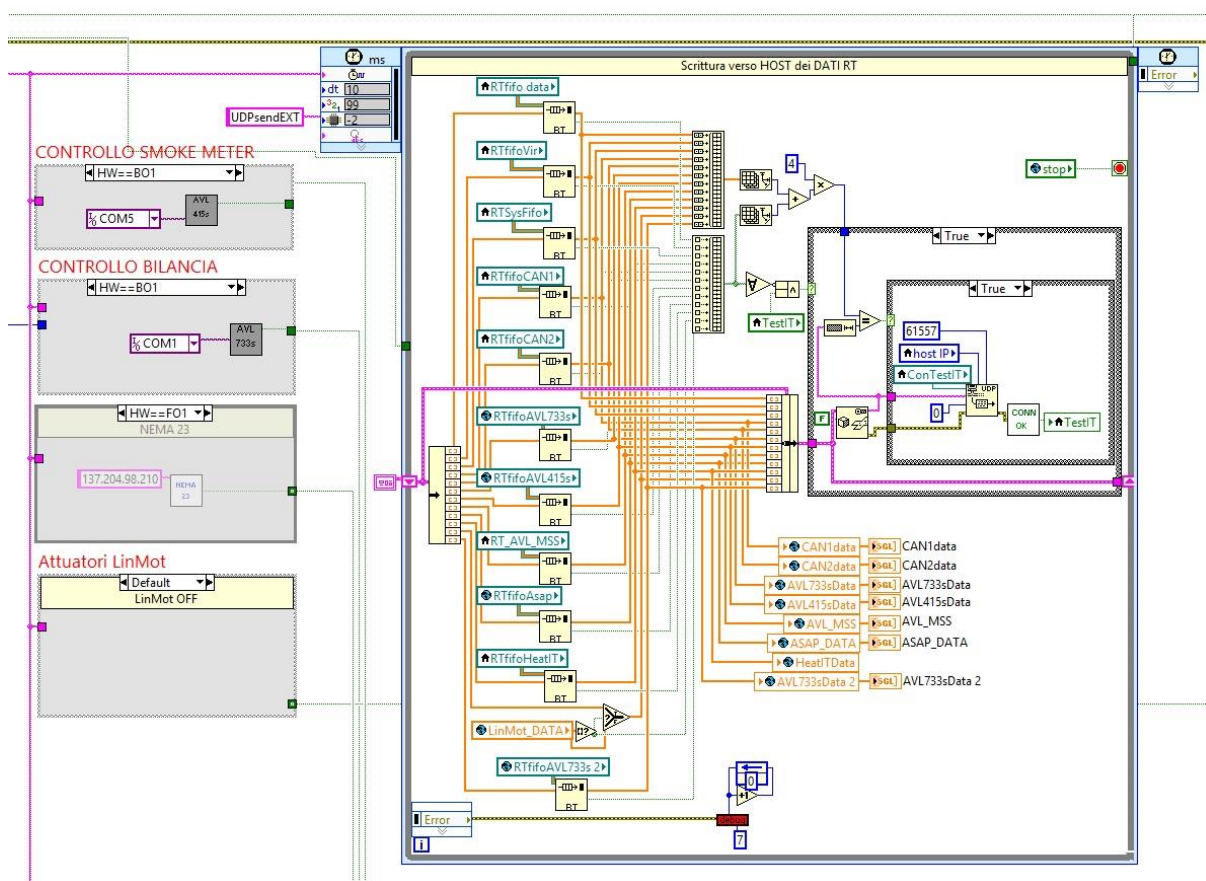


Figura 57 Particolare di rMain relativo alla comunicazione UDP

Allo stesso modo rMain riceve da Host un **cluster** di array che viene spaccettato (*unbundle*) e suddiviso nei vari subVI presenti al suo interno.

7 – Appendice

In quest'ultimo capitolo vengono riportati i più importanti lavori svolti per ottimizzare la disposizione dell'hardware in sala.

Obiettivo del lavoro di ottimizzazione è la composizione di un rack industriale da porre direttamente in cella, all'interno del quale si installa tutta la componentistica hardware necessaria.

La scelta del rack è ricaduta sull'armadio da 19 pollici serie **ODN** acquistato dall'azienda **O. DE NICOLAI s.r.l.** di Milano.

L'armadio ha le seguenti caratteristiche:

- Kit 4 ventole per il raffreddamento;
- Kit 4 ruote girevoli;
- Kit 4 piedini livellamento;
- 4 montanti numerati.

L'interno è accessibile tramite una porta frontale in vetro con serratura a chiave, ante laterali asportabili, una porta retro chiusa, apribile con serratura a chiave. L'armadio è completamente smontabile.

L'ingombro è standard per una larghezza di 600 mm e una profondità di 800 mm. Le alternative erano a profondità 600 mm o 1000 mm; si è ricorsi all'800 mm sia per un compromesso fra ingombri e costi che in quanto lo spazio da occupare era sufficiente.

I dispositivi sono collocati all'interno grazie a guide forate progettate secondo le specifiche **EIA-310** (EIA: *Electronic Industries Alliance*) che prevedono alloggiamenti in Unità Rack. L'unità corrisponde a un'altezza di 1,75 pollici o 44.45 mm per una larghezza di 48,5 mm.

Pannelli e dispositivi sono quindi progettati per essere attaccati sulle guide secondo dimensioni 1U, 2U o 4U (eventualmente half-rack).

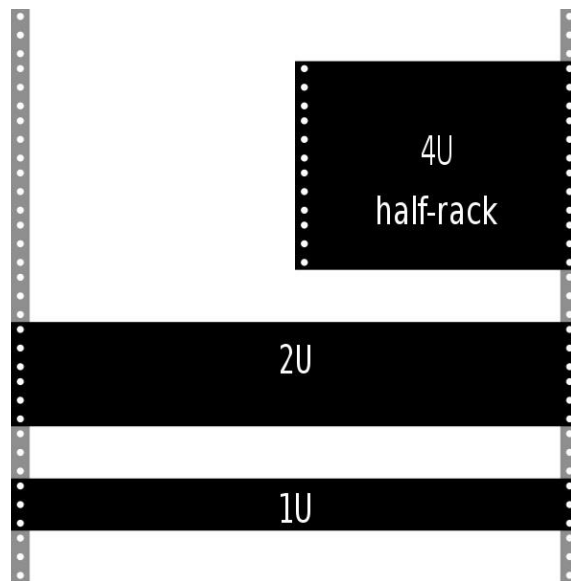


Figura 58 Rappresentazione delle unità rack

Nel lavoro di ottimizzazione era prevista anche la progettazione di due pannelli forati di dimensione 1U per supportare tutte le termocoppie presenti in sala. Il progetto è disegnato con **SolidWorks**:

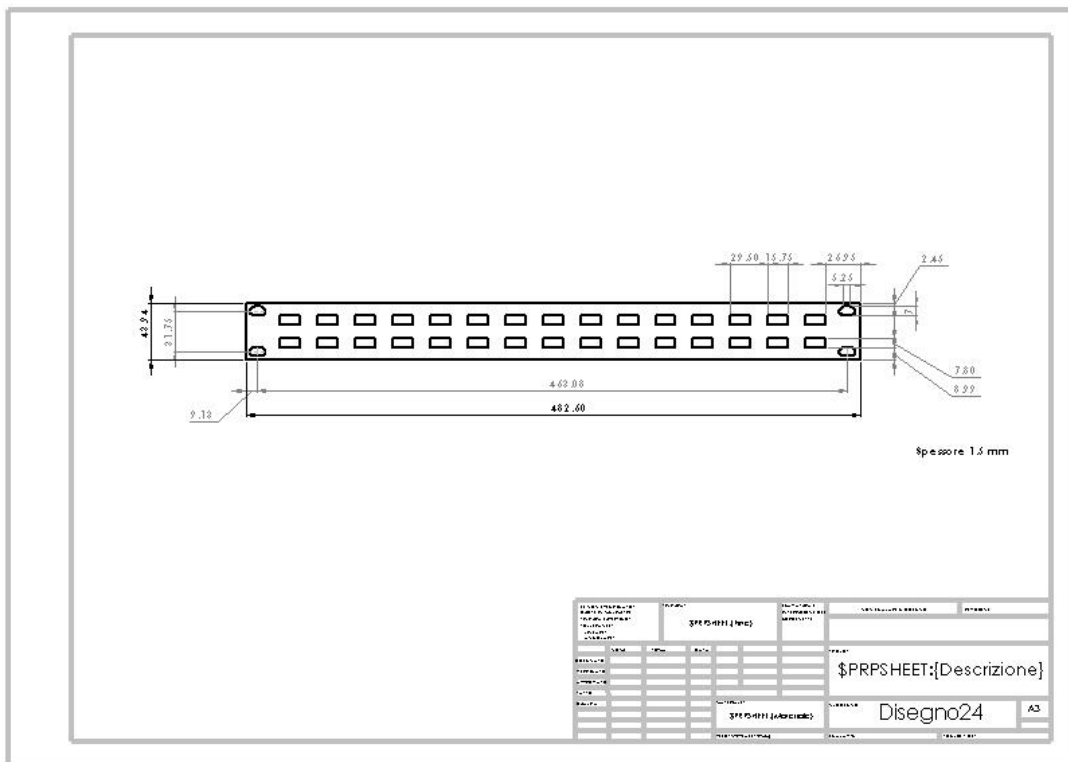


Figura 59 Progetto in Solidwork del pannello per le termocoppie

Per quanto riguarda l'hardware da collocare all'interno del rack invece:

- Pc Real-Time;
- PXI 1033 con le relative schede;
- BNC 2110 e BNC 2090 per segnali input ed output;
- Un pannello relay per le attuazioni;
- SCXI 1000 per le termocoppie;
- MXI express-Rio 9154 con i relativi moduli;
- SAT 2000 relativo al freno motore;
- Guide Din con l'alimentazione.

Anche in questo caso l'ottimizzazione è avvenuta tramite disegni in Solidworks per la ricerca della disposizione ottimale.

In conclusione della tesi viene riportato il risultato dell'ottimizzazione sia nel disegno progettuale, sia nella realtà:



Figura 60 Confronto fra progetto e risultato

Bibliografia

- Manuale Operativo AVL 733S;
- Manuale Operativo AVL 415S;
- Manuale Operativo MDrive 23 Plus;
- Manuale TestIT.