



**Giovanni Schgör (g.schgor)**

# Strumentazione e controllo virtuali - Parte prima

4 June 2006

Articolo n° 1 su 3 del corso "[Strumentazione e controllo virtuali](#)".  
Vai all'[indice](#) del corso.

Paragrafi dell'articolo:

1. [Sommario](#)
2. [L'origine della strumentazione virtuale](#)
3. [USB-6008 e LabVIEW](#)
4. [Esempio di applicazione \(logica combinatoria\)](#)
5. [Il Set-Reset flip-flop](#)
6. [Il temporizzatore](#)
7. [Il "mini PLC"](#)

Strumentazione e controllo virtuali

G. Schgör

Sommario

Questa serie di articoli vuole essere una semplice introduzione in una delle tecnologie destinate a cambiare le soluzioni di automazione industriale dagli attuali apparati specifici a configurazioni standardizzate, basate su hardware unificato e specializzate mediante software.

L'evoluzione degli attuali PLC (Programmable Logic Controller) e dei sistemi di controllo computerizzati tende infatti a fondere in pochi tipi di apparati sia le funzioni di logica industriale che quelle di acquisizione, di elaborazione, di controllo e di supervisione di tutte le grandezze in gioco in un impianto.

La scelta di un piccolo (ed economico) apparato per affrontare il tema è dettata dall'esigenza di mantenere semplici le illustrazioni dei vari esempi applicativi. Si fa qui infatti riferimento al dispositivo USB-6008 della National Instruments, che ha possibilità di gestire 12 fra ingressi

ed uscite digitali, 8 ingressi e 2 uscite di tipo analogico, ed è collegabile ad un PC mediante porta USB.

Il linguaggio di programmazione è LabVIEW, di tipo grafico, con blocchi funzionali già disponibili e che devono essere opportunamente personalizzati e collegati fra loro.

Una prima parte sarà dedicata alla realizzazione di un "mini PLC", con l'utilizzo dei segnali digitali, mentre le successive affronteranno rispettivamente la visualizzazione di dati analogici (oscilloscopio virtuale), la generazione di forme d'onda ed il loro utilizzo in esperimenti di controllo (regolazione automatica), l'elaborazione dei segnali e così via.

E' evidente lo scopo didattico di tale presentazione, che non pretende certo di esaurire la molteplicità delle possibili applicazioni, né di sostituire i corsi specializzati organizzati dalla stessa National Instruments, ma solo di fornire una informazione, più semplice possibile, delle potenzialità insite in questa tecnologia.

### L'origine della strumentazione virtuale

La National Instruments (NI) è nata 30 anni fa sulla semplice idea di utilizzare gli allora nascenti minicomputer per acquisire ed elaborare i dati di misura al posto della tradizionale strumentazione (creando appunto quella "virtuale"). Attualmente è fra le prime 100 industrie degli Stati Uniti.

Con l'avvento del PC, NI ha sviluppato un particolare software grafico, il LabVIEW, che facilita la programmazione delle applicazioni attraverso l'uso di una vastissima serie di 'blocchi funzionali'.

Questo evidentemente rende rapida la preparazione di soluzioni personalizzate, ma richiede una conoscenza approfondita delle varie possibilità e delle modalità di utilizzo. Non è quindi sempre facile per il principiante affrontare tale compito e scopo di questi articoli è appunto quello di semplificare tale approccio.

NI, partendo dalla sua esperienza strumentistica, oltre ad aver sviluppato una vasta gamma di apparati di 'interfaccia' (schede di condizionamento segnali, di conversione A/D e D/A, di isolamento, ecc.), propone ora 'soluzioni integrate' di automazione industriale, incorporando configurazioni di microprocessori che possono essere programmate con lo stesso linguaggio LabVIEW, mediante memorie

da inserire nelle configurazioni stesse (FPGA : Field-Programmable Gate Array)

Si ha così una 'intelligenza distribuita' che è in linea con i più recenti criteri di controllo degli impianti industriali.

USB-6008 e LabVIEW

Questo dispositivo è tra i più semplici fra la vastissima gamma dei prodotti NI ed è costituito da interfacce di segnali sia digitali che analogici verso un PC, mediante porta USB (Universal Serial Bus).



La figura mostra tale componente, evidenziando le morsettiere laterali dei 12 segnali digitali (a livelli TTL), degli 8 ingressi analogici (+/- 10V) e delle 2 uscite analogiche (0-5 V).

Non è mostrato il cavo USB (collegato al connettore in alto).

I segnali digitali possono essere programmati per essere utilizzati come ingressi oppure come uscite, e vi è inoltre un ingresso di conteggio (32 bits).

Nel nostro caso, si è scelto di utilizzare 8 ingressi (port 0) e 4 uscite (port1).

Tutti i dettagli delle caratteristiche sono comunque riportati in "User Guide and Specifications", disponibile in Internet: <http://www.ni.com/pdf/manuals/371303e.pdf>

NI offre una versione per studenti (student kit) che comprende anche il software LabVIEW, in modo che l'USB-6008 possa essere programmato liberamente in tale linguaggio.

E' quello che seguiremo, dopo una breve conoscenza delle sue caratteristiche (ma senza entrare nei dettagli operativi, descritti nei manuali).

Lanciato LabVIEW, compare una prima finestra **Front Panel**, che rappresenta il pannello operativo in cui si possono installare oggetti quali pulsanti, lampade, selettori, strumenti indicatori o registratori, ecc.

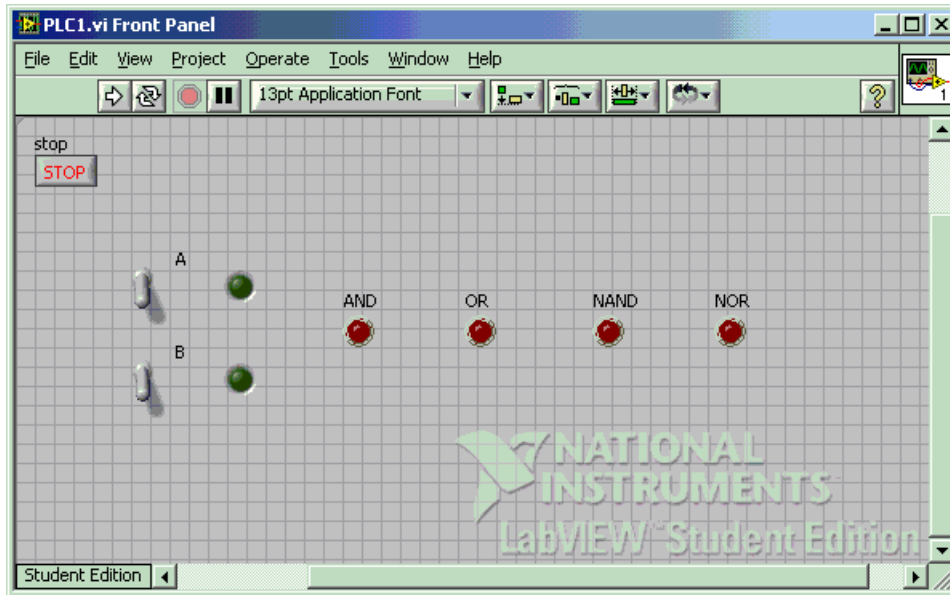
Dal Front Panel è infatti possibile richiamare altre finestre, quali **Controls palette** (che contiene i vari oggetti disponibili) e **Block Diagram** (destinato a contenere il programma, in forma grafica).

Dal Block Diagram è possibile richiamare **Functions Palette** (che contiene le icone simboliche delle varie funzioni di programmazione) e **Tools Palette** (che contiene le modalità di collegamento e di azione sulle icone utilizzate).

Tutta la programmazione consiste nella gestione di queste 5 finestre al fine di costruire nel Block Diagram il grafico delle funzioni richiamate, per far funzionare opportunamente, in fase di esecuzione, gli apparati inseriti nel Front Panel.

Esempio di applicazione (logica combinatoria)

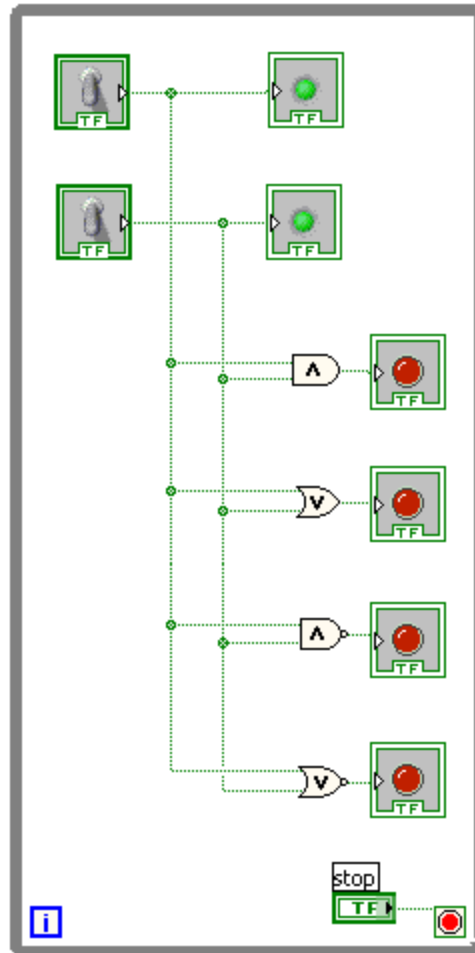
Vediamone subito un esempio, utilizzando segnali digitali creati internamente (per il momento non utilizziamo l'USB-6008): nel Front Panel disponiamo 2 interruttori a levetta (A e B) e dei LED di segnalazione (verdi per la visualizzazione degli stati rispettivamente di A e B, e rossi in corrispondenza di varie funzioni logiche AND, OR, NAND e NOR). Ovviamente tutti questi "componenti" sono "virtuali", scelti nella finestra Controls Palette.



Queste scelte comportano però che nella finestra Block Diagram compaiano gli equivalenti funzionali di tali componenti, e in questa dovremo quindi eseguire gli opportuni collegamenti (cablaggi virtuali) inserendo, ove richiesto, le funzioni logiche desiderate (ricavate dal Functions Panel).

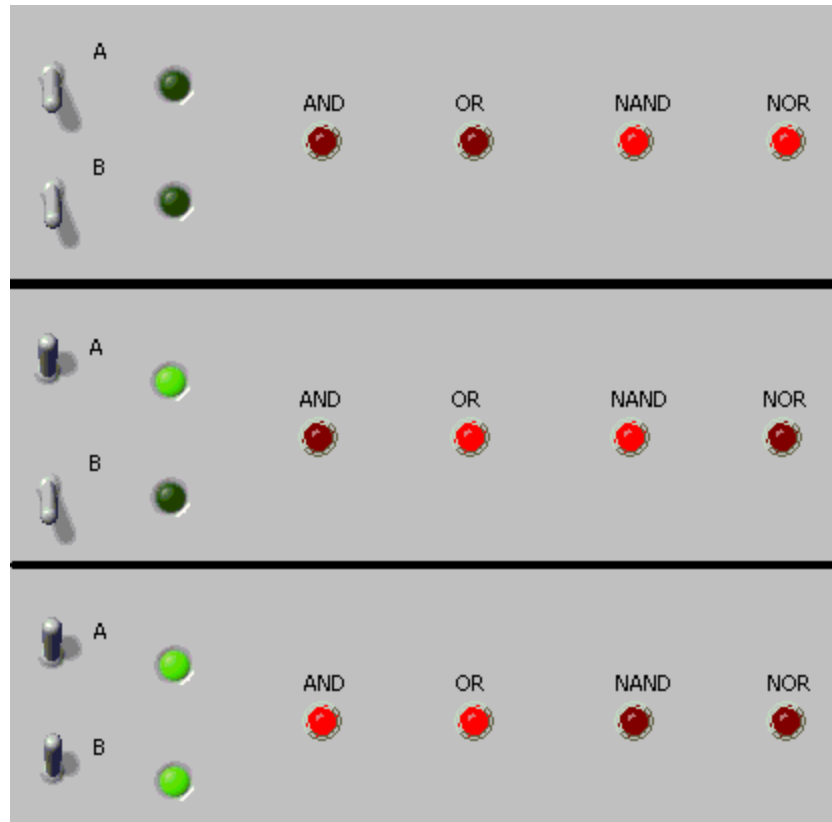
La figura seguente mostra il Block Diagram completato (si notino i simboli convenzionali delle funzioni logiche inserite fra gli interruttori ed i LED rossi).

Si noti anche il riquadro in neretto che circonda lo schema: non è un abbellimento, ma il simbolo di un loop di programma, che corrisponde a "while", e che ha lo scopo di far ripetere il programma in tempo reale, finché non viene arrestato col pulsante di **stop** nel Front Panel.



Siamo ora pronti a far funzionare il nostro programma. Infatti se nel Front Panel attiviamo col mouse la freccetta bianca, lo mandiamo in esecuzione e da quel momento tutte le azioni che faremo (sempre col mouse) sulle levette degli interruttori, si trasformeranno in comandi di accensione o spegnimento delle lampade, secondo la "logica" predisposta.

La figura seguente mostra il risultato "fotografico" di 3 possibili combinazioni dei segnali A e B, da cui può essere rilevata la corrispondenza alle singole funzioni logiche utilizzate.. (la quarta possibile combinazione ripete quella di mezzo, scambiando solo A con B).

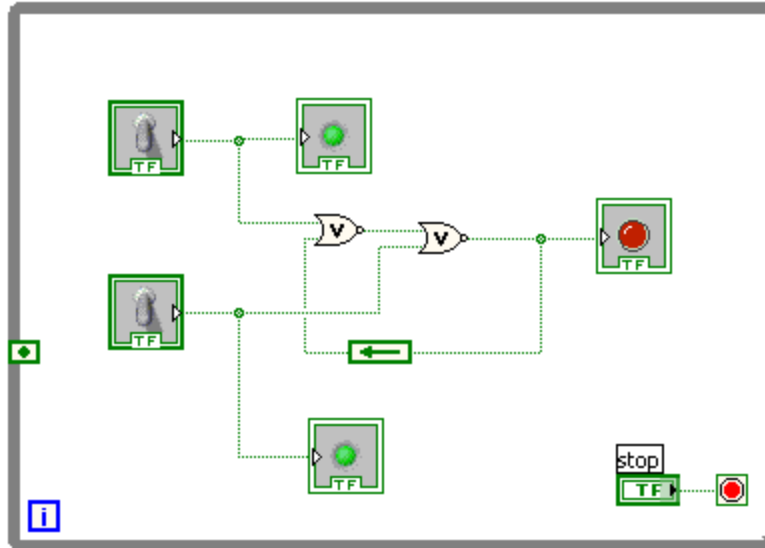


Non occorre essere degli esperti in Logica Booleana per capire che già solo con queste funzioni, qualsiasi circuito combinatorio può essere realizzato (in realtà ve ne sono disponibili altre, che qui tralasciamo per semplicità).

Nell'automazione non ci si può però limitare alla logica combinatoria. Sappiamo che sono indispensabili anche elementi sequenziali: almeno un circuito di memorizzazione ed uno di temporizzazione. E' quello che vedremo nei prossimi esempi.

## Il Set-Reset Flip-Flop

Volendo costruire un classico circuito di memorizzazione, disponiamo nel Front Panel 2 interruttori (S ed R) con le rispettive lampade di stato (verdi), ed una lampada (rossa) per il segnale d'uscita (Q). Il programma, nel Block Diagram, sarà allora:

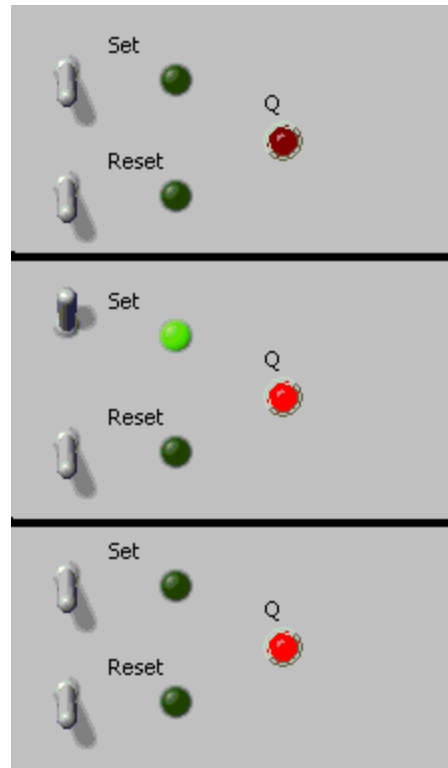


in cui si nota un rettangolino con freccia all'indietro: questo è necessario per indicare che vi è un segnale di "retroazione", il che fa diventare il circuito da combinatorio a sequenziale.

Il risultato "fotografico" (come nel caso precedente) della sequenza del segnale di Set (S), dimostra che l'uscita Q rimane accesa anche quando S torna a zero (corrispondente lampada verde spenta).

Ovviamente Q si spegnerà solo se interviene R (Reset).





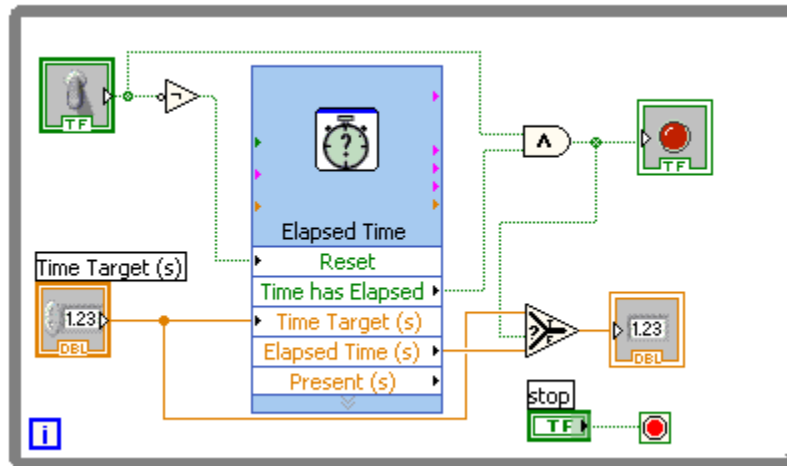
## Il temporizzatore

Per completare la serie di strumenti di base necessari per realizzare una logica industriale, occorre aggiungere un circuito di ritardo prefissabile ( ad es. in decimi di s). Vediamo allora come procedere con LabVIEW.

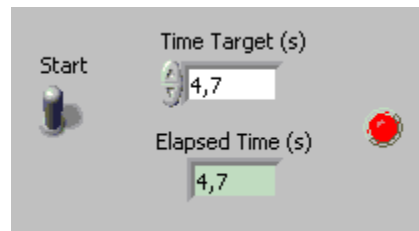


Nel Front Panel verranno sistemati l'interruttore di start e la lampada di segnalazione. Ove sia richiesto di poter variare il tempo di ritardo, occorre aggiungere l'impostazione (Time Target) e, volendo, anche un indicatore del tempo trascorso (Time Elapsed).

Nel Block Diagram, dovrà invece essere sistemata la funzione di temporizzazione ed i collegamenti necessari. (si noti il deviatore all'ingresso dell'indicatore, che ha la funzione di limitare l'indicazione stessa al valore impostato).



Il risultato finale, dopo che si sia impostato il tempo voluto e si sia sollevato l'interruttore, è l'accensione della luce rossa dopo trascorso esattamente questo tempo.



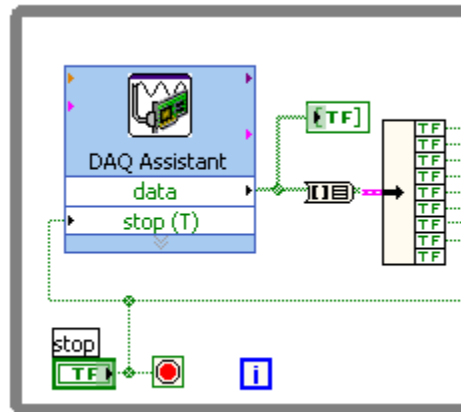
Con questi semplici circuiti siamo ora in grado di risolvere qualsiasi problema posto dalla normale logica industriale, ma ci manca ovviamente "l'interfacciamento" del nostro programma con i segnali esterni. Vedremo quindi ora la gestione dell'USB-6008.

## Il "mini PLC"

Si è detto che utilizzeremo l'USB-6008 prevedendo 8 ingressi digitali (port0) e 4 uscite digitali (la metà inferiore

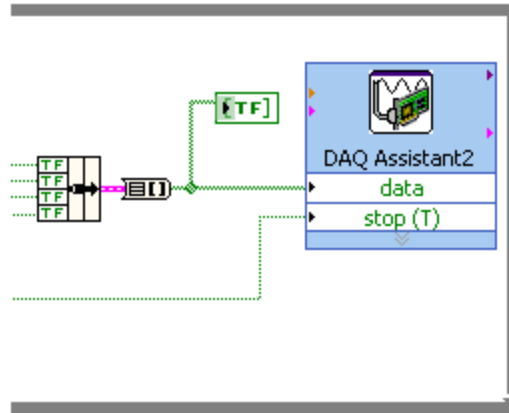
del port1). Si tratta allora di utilizzare in LabVIEW dei moduli di software personalizzati per ricavare dall'USB-6008 i segnali booleani corrispondenti a ciascun singolo ingresso od uscita.

Nel Block Diagram, la programmazione per ottenere gli ingressi è:



in cui il modulo di acquisizione dati (DAQ Assistant) è programmato per ingressi digitali (single line). Il dato in uscita da questo blocco, viene applicato ad un convertitore (Array To Cluster), per ricavarne alla fine i singoli segnali booleani (T F, cioè True/False, con una funzione chiamata "unbundle"). Un indicatore ad 8 LED verdi ( [TF] ) mostra sul Front Panel lo stato dei singoli ingressi

Per le uscite, praticamente si fa il percorso inverso.



E' ovvio che il loop "while" è lo stesso per le due funzioni di lettura stati degli ingressi e scrittura degli stati delle uscite. In mezzo a questi, devono chiaramente essere sviluppati i circuiti logici necessari al particolare controllo, utilizzando (come già visto) le funzioni logiche della Functions Palette.

Non dovrebbe essere necessario sottolineare che, una volta superate le difficoltà dell'approccio, la realizzazione di configurazioni logiche di controllo diventa estremamente semplice.